intel®

APPLICATION NOTE

AP-15

8255 Programmable Peripheral Interface Applications

Alan Ebright
Microcomputer Applications

PRICE $1.00

8255 Programmable Peripheral Interface Applications

Contents

Intel Corporation assumes no responsibility for the use of any circuitry or software other than circuitry or software embodied in an Intel product. No other circuit patent licenses are implied.

Related Intel Publications

"Intel 8080 Microcomputer Systems User's Manual"

"Memory Design Handbook"

"Using the 8251 Universal Synchronous/Asynchronous Receiver/Transmitter"

INTRODUCTION

Microprocessor-based system designs are a cost-effective solution to a wide variety of problems. When a system designer is presented with the task of selecting a microprocessor for a design, the capabilities of the microprocessor should not be the only consideration. The microprocessor should be an element of a compatible family of devices. The MCS-80 component family is a group of compatible devices which have been designed to directly address and solve the problems of microprocessor-based system design. One member of the MCS-80 component family is Intel's 8255 programmable peripheral interface chip. This device replaces a significant percentage of the logic required to support a variety of byte oriented Input/Output interfaces. Through the use of the 8255, the I/O interface design task is significantly simplified, the design flexibility is increased, and the number of components required is reduced.

This application note presents detailed design examples from both the hardware and software points of view. Since the 8255 is an extremely flexible device, it is impossible to list all of the applications and configurations of the device. A number of designs are presented which may be modified to fulfill specific user interface requirements.

Detailed design examples are discussed within the context of the 8080 system shown in Figure 1. The basic 8080 system is composed of the CPU module, memory module, and the I/O module. CPU module and memory module design are discussed within other Intel publications. This application note deals exclusively with I/O module design.

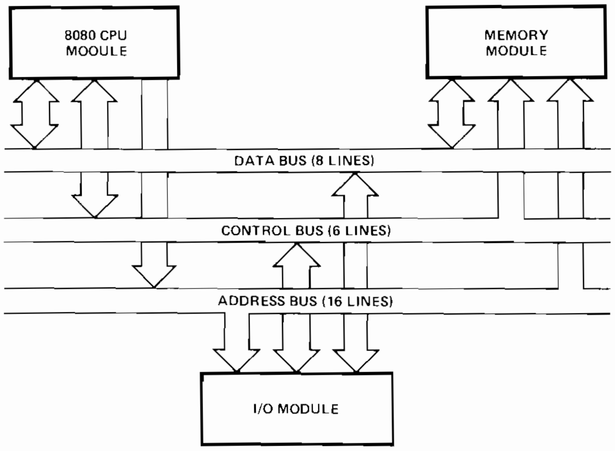


Figure 1. Typical 8080 System

It is assumed that the reader is familiar with the "8080 Microcomputer Systems User's Manual", particularly the 8255 device description.

OVERVIEW OF THE 8255

The 8255 block diagram shown in Figure 2 has been divided into three sections: 8080 CPU Module Interface, Peripheral Interface, and the Internal Logic.

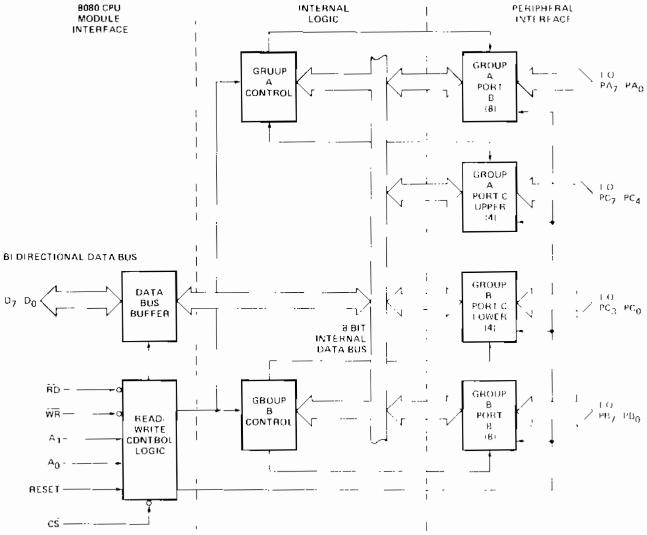


Figure 2. 8255 Block Diagram

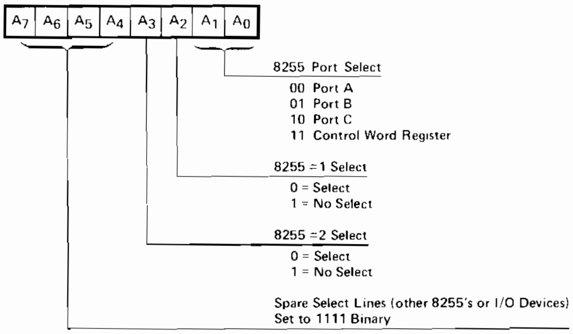
8080 CPU MODULE INTERFACE

The 8255 is a compatible member of the MCS-80 component family and, therefore, may be directly interfaced to the 8080. Figure 3 displays one method of interconnecting the 8255 and an 8080 CPU module. The 8080 CPU module consists of the 8080A CPU, the 8224 Clock Generator, and the 8228 System Controller. The system shown in Figure 3 utilizes a linear select scheme which dedicates an address line as an exclusive enable (chip select) for each specific I/O device. The chip select signal is used to enable communication between the selected 8255 and the 8080 CPU. I/O Ports A, B, C or the Control Word Register are selected by the two port select signals (A_1, A_0). These signals (A_1 and A_0) are driven by the least significant bits of the address bus. The I/O port select characters required by this configuration are shown in Figure 4.

When a system utilizing the linear select scheme is implemented, a maximum of six I/O devices may be selected. If more than six I/O devices must be addressed, the six device select bits must be encoded to generate a maximum of 64 device select lines. Note that when large systems are implemented, bus loading considerations may require that bus drivers be included in the CPU module. The MCS-80 component family contains parts which are designed to perform this function (8216, 8226).

The 8255 I/O read ($\overline{RD}$) and I/O write ($\overline{WR}$) signals may be directly driven by the 8228. This results in an isolated I/O architecture where 8080 Input/Output instructions are used to reference an independent I/O address space. An alternate approach is memory mapped I/O. This architecture treats an area of memory as the I/O address space. The memory mapped I/O architecture utilizes 8080 memory reference instructions to access the I/O address space. Interfacing with the 8080 is outlined in Chapter 3 of the "8080 Microcomputer User's Manual".

The most important feature of the 8255 to 8080 CPU Module Interface is that for small system designs the 8255 may be interfaced directly to the standard MCS-80 component family with no external logic. Minimum external logic is required in large system designs.



Port Selected	Hexadecimal Port Select Character (Used with IN or OUT Instructions)
Port A 8255 =1	F8
Port B 8255 =1	F9
Port C 8255 =1	FA
Control Word Register 8255 =1	FB
Port A 8255 =2	F4
Port B 8255 =2	F5
Port C 8255 =2	F6
Control Word Register 8255 =2	F7

Figure 4. I/O Port Select Characters

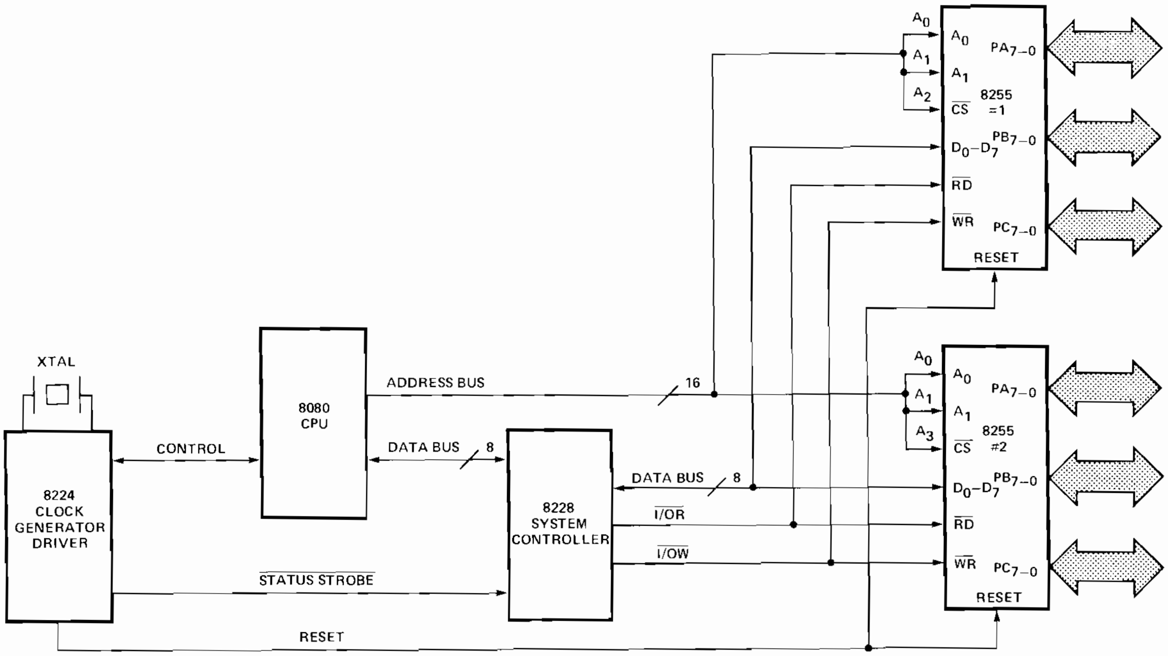


Figure 3. Linear Select 8255 Interconnect

PERIPHERAL INTERFACE SECTION

The peripheral interface section contains 24 peripheral interface lines, buffers, and control logic. The characteristics and functions of the interface lines are determined by the operating mode selected under program control. The flexibility of the 8255 is due to the fact that the device is programmable. Three modes of operation may be selected under program control: Mode 0 – Basic Input/Output, Mode 1 – Strobed Input/Output with interrupt support, and Mode 2 – Bidirectional bus with interrupt support. Through selecting the correct operating mode, the interface lines may be configured to fulfill specific interface requirements. The characteristics of the interface lines within each mode must be understood so that the designer may utilize the 8255 to achieve the most efficient design. Table I lists the basic features of the peripheral interface lines within each mode group. Figure 5 shows the grouping of the peripheral interface lines within each mode.

Table I. Features of Peripheral Interface Lines

Mode 0 – Basic Input/Output
Two 8-bit ports Two 4-bit ports with bit set/reset capability Outputs are latched Inputs are not latched
Mode 1 – Strobed Input/Output
One or two strobed ports Each Mode 1 port contains: 8-bit data port 3 control lines Interrupt support logic Any port may be input or output If one Mode 1 port is used, the remaining 13 lines may be configured in Mode 0. If two Mode 1 ports are used, the remaining 2 bits may be input or output with bit set/reset capability.
Mode 2 – Strobed Bidirectional Bus
One bidirectional bus which contains: 8-bit bidirectional bus supported by Port A 5 control lines Interrupt support logic Inputs and outputs are latched The remaining 11 lines may be configured in either Mode 0 or Mode 1.

One feature of Port C is important to note. Each Port C bit may be individually set and reset. Through the use of this feature, device strobes may be easily generated by software without utilizing external logic. The Mode 1 and Mode 2 configurations use a number of the Port C lines for interrupt control lines. Thus, the 8255 contains a large portion of the logic required to implement an interrupt driven I/O interface. This feature simplifies interrupt driven hardware design and saves a significant amount of the external logic that is normally required when less powerful I/O chips are used. In fact, the design examples contained in this application note describe how interrupt driven interfaces may be designed such that the only interrupt control logic required is that contained in the 8255.

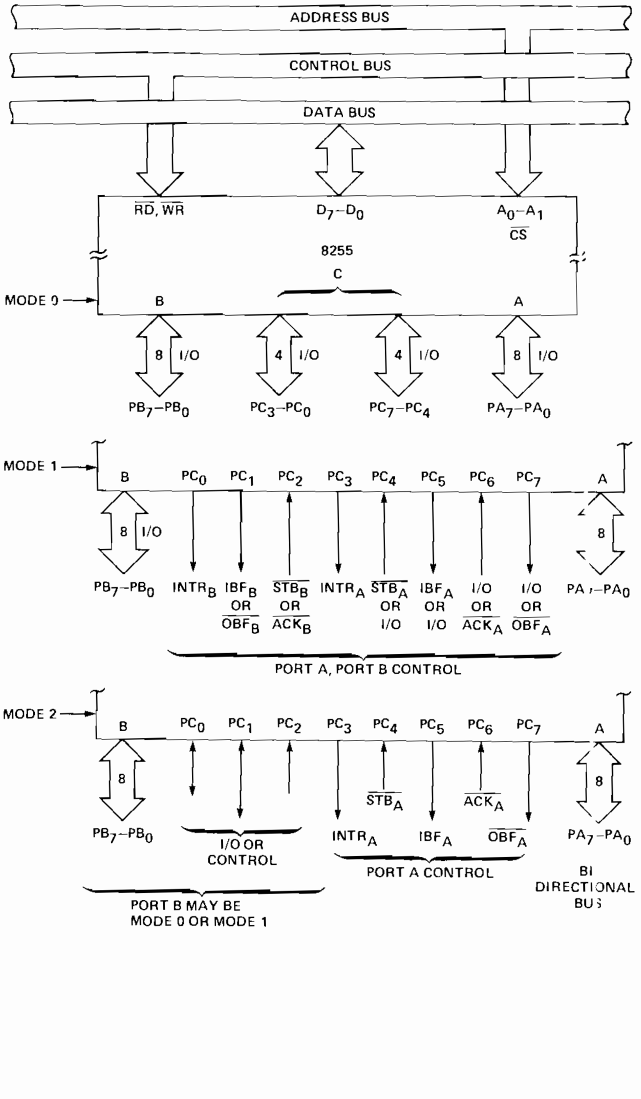


Figure 5. Grouping of Peripheral Interface Lines

INTERNAL LOGIC SECTION

The internal logic section manages the transfer of data and control information on the internal data bus (refer to Figure 2). If the port select lines (A_1 and A_0) specify Ports A, B, or C, the operation is an I/O port data transfer. The internal logic will select the specified I/O port and perform the data transfer between the I/O port and the CPU interface. As was previously mentioned, both the functional configuration of each port and bit set/reset on Port C are controlled by the system's software. When the control word register is selected, the internal logic performs the operation described by the control word. The control word contains an opcode field which defines which of the two functions are to be performed (mode definition or bit set/reset).

Mode Definition

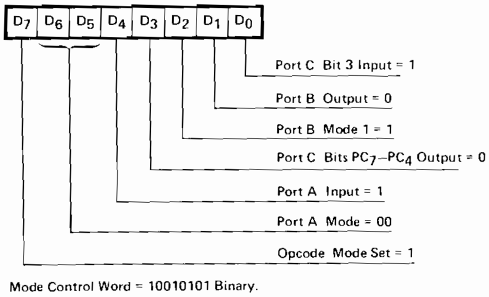
When the opcode field (Bit 7) of the control word is equal to a one, the control word is interpreted by the 8255 as a mode definition control word. The mode definition control word (shown in Figure 6) is used to specify the configuration of the 24 8255 peripheral interface lines. The system's software may specify the modes of Port A and Port B independently. Port C may be treated independently or divided into two portions as required by the Port A and Port B mode definitions.

Example #1: This example demonstrates how a mode control word is constructed and issued to an 8255. The mode control word is passed to the device through the use of an output instruction that references an 8080 I/O port address. The value of the I/O port address is determined by the 8080 CPU interface implemented. This example references the I/O port addresses realized by the simple 8080 to 8255 interface shown in Figure 3.

If an 8255 is to be configured through the use of the mode control word interface as:

Port A	Mode 0 Input
Port B	Mode 1 Output
Port C	Bits PC_7–PC_4 Output
Port C	Bit 3 Input

The following mode control word is used:



D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
Port C Bit 3 Input = 1
Port B Output = 0
Port B Mode 1 = 1
Port C Bits PC_7–PC_4 Output = 0
Port A Input = 1
Port A Mode = 00
Opcode Mode Set = 1
Mode Control Word = 10010101 Binary.

The assembly language program is:

```
CWR          EQU    0FBH          ;8255 #1 CONTROL WORD REGISTER
;*****
;       ISSUE MODE CONTROL WORD
;*****
             MVI    A,10010101B   ;GET MODE CONTROL WORD
             OUT    CWR           ;OUTPUT TO 8255 #1 CONTROL WORD
                                  ;REGISTER
```

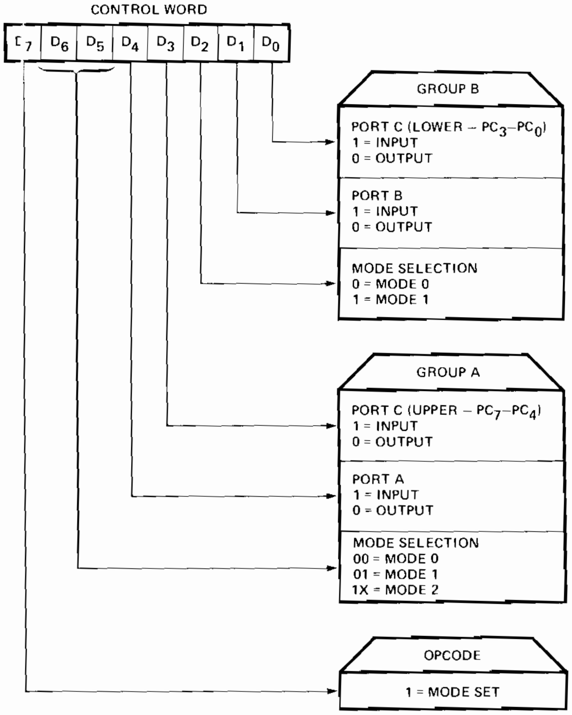


Figure 6. Mode Definition Control Word

Bit Set/Reset

When the opcode field (Bit 7) of the control word is equal to a zero, the control word is interpreted by the 8255 as a Port C bit set/reset command word (see Figure 7). Through the use of the bit set/reset command, any of the 8 bits on Port C may be independently set or reset. Note that control word bits 6–4 are not used. Bits 6–4 should be set to zero.

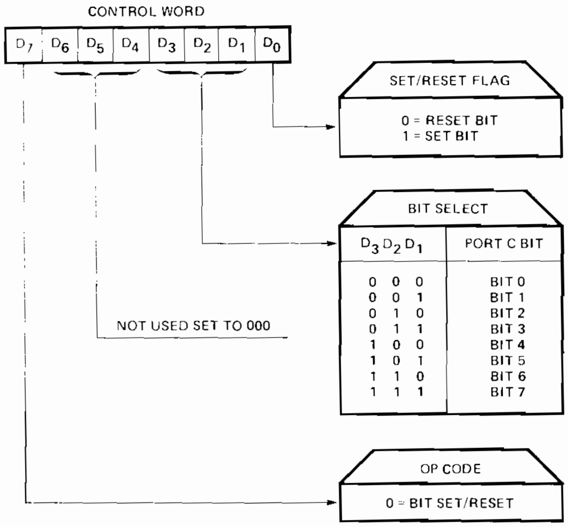
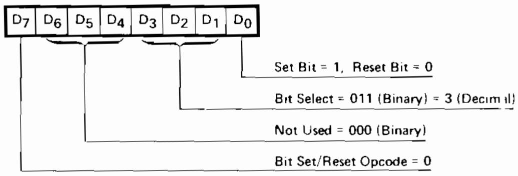


Figure 7. Bit Set/Reset Control Word

Example #2: This example demonstrates how a Port C bit set/reset control word is constructed and issued to an 8255. The bit set/reset control word is passed to the device through the use of an output instruction that references an 8080 I/O port address. The value of the I/O port address is determined by the 8080 CPU interface implemented. This example references the I/O port addresses realized by the simple 8080 to 8255 interface shown in Figure 3.

Control word (see Figure 7).


D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
Set Bit = 1, Reset Bit = 0
Bit Select = 011 (Binary) = 3 (Decimal)
Not Used = 000 (Binary)
Bit Set/Reset Opcode = 0

The control word for set Port C bit 3 is 00000111 binary.
The control word for reset Port C bit 3 is 00000110 binary.

The assembly language program is:

```
CWR     EOU     0FBH              ; 8255 #1 CONTROL WORD REGISTER
; *****
;       SET BIT 3
; *****
        MVI     A, 00000111B      ; GET SET BIT 3 CONTROL WORD
        OUT     CWR               ; OUTPUT TO 8255 #1 CONTROL WORD REGISTER
; *****
;       RESET BIT 3
; *****
        MVI     A, 00000110B      ; GET RESET BIT 3 CONTROL WORD
        OUT     CWR               ; OUTPUT TO 8255 #1 CONTROL WORD REGISTER
```

NOTE: An MVI instruction is used to load the reset bit 3 control word into the A register. Since it is known that the set bit control word is already in the A register, a "DCR A" Instruction could be used to generate the correct control word and save one byte of code.

00000111 – 1 = 00000110 (RESET BIT 3 CONTROL WORD)

Example #3: This example demonstrates one simple method of performing a bit set/reset operation on Ports A and B. The state of any output port may be determined by reading the port. The assembly language program which may be used to set/reset Port A or B bits is:

```
PORTA   EOU     0F8H              ; 8255 #1 PORT A
; *****
;       SET BIT 0
; *****
        IN      PORTA             ; GET STATE OF PORT
        ORI     01H               , SET BIT 0
        OUT     PORTA             ; OUTPUT TO PORT
; *****
;       RESET BIT 0
; *****
        IN      PORTA             ; GET STATE OF PORT
        ANI     0FEH              ; RESET BIT 0
        OUT     PORTA             ; OUTPUT TO PORT
```

INTERRUPT CONTROL LOGIC STATUS WORDS

As previously mentioned, the 8255 Mode 1 and Mode 2 configurations support interrupt control logic. If a read of Port C is issued when the 8255 is configured in Mode 1, the software will receive the Mode 1 status word shown in Figure 8. The bits in the status word correspond to the state of the associated Port C lines (buffer full, interrupt request, etc.). The INTE bit shown in the status word corresponds to the interrupt enable flip-flop contained in the 8255. This signal is not available externally. The structure of the Mode 1 status word varies as a function of the mode of the 8255. Example #4 shows the status word which results from reading Port C from an 8255 which is configured with Port A Mode 1 input and Port B Mode 1 output.

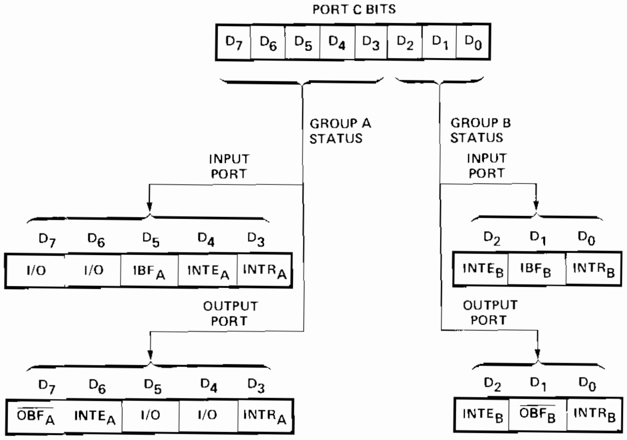


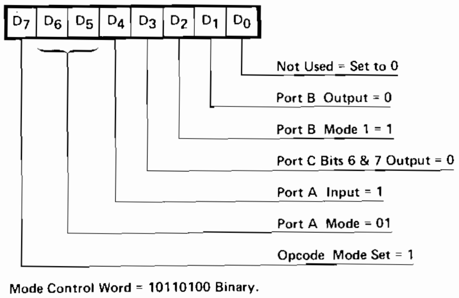
Figure 8. Mode 1 Status Word

Example #4 – MODE 1 STATUS WORD

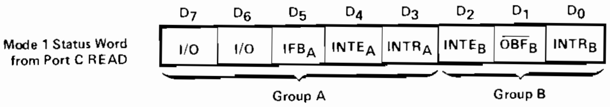
If an 8255 is to be configured through the use of the mode control word interface as:

Port A	Mode 1 Input
Port B	Mode 1 Output
Port C	Bits 6 & 7 Output

The following mode control word is used:



After the 8255 mode control word has been issued, a READ of Port C will obtain the following Mode 1 status word:


Mode 1 Status Word from Port C READ
D7 D6 D5 D4 D3 D2 D1 D0
I/O | I/O | IFB_A | INTE_A | INTR_A | INTE_B | OBF_B | INTR_B
Group A
Group B

NOTE: The Port C I/O bits D_7 and D_6 should be modified through the use of the Port C bit set/reset command word. If a write to Port C is issued, the $INTE_A$ and $INTE_B$ bits may be inadvertently modified by the user. The IBF_A, $INTR_A$, $\overline{OBF}_B$, and $INTR_B$ bits will not be modified by either a write to Port C or a bit set/reset command. These four bits always reflect the state of the interrupt control logic.

Note that the Mode 2 status word (shown in Figure 9) differs from the Mode 1 status word. The format of the status word data bits D_2–D_0 are defined by the specification of the Port B configuration. Example #5 shows the structure of the Mode 2 status word when the 8255 is configured with Port A Mode 2 (bidirectional bus) and Port B Mode 1 input.

The Mode 1 and Mode 2 status words reflect the state of the interrrupt logic supported by the 8255.

Example #6 demonstrates how the interrupt enable bits are controlled through the use of the Port C bit set/reset feature. The application examples provide a more detailed explanation of the use of the Port C status word in the Mode 1 and Mode 2 configurations.

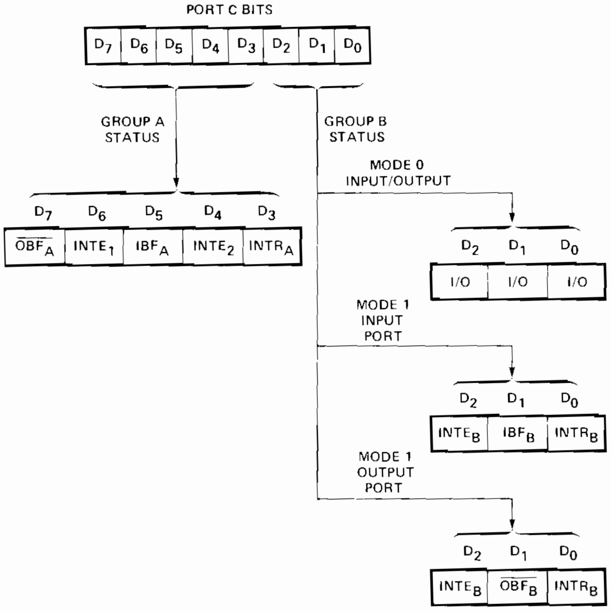

PORT C BITS
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
GROUP A STATUS
GROUP B STATUS
D_7 D_6 D_5 D_4 D_3
$\overline{OBF}_A$ | $INTE_1$ | IBF_A | $INTE_2$ | $INTR_A$
MODE 0 INPUT/OUTPUT
D_2 D_1 D_0
I/O | I/O | I/O
MODE 1 INPUT PORT
D_2 D_1 D_0
$INTE_B$ | IBF_B | $INTR_B$
MODE 1 OUTPUT PORT
D_2 D_1 D_0
$INTE_B$ | $\overline{OBF}_B$ | $INTR_B$

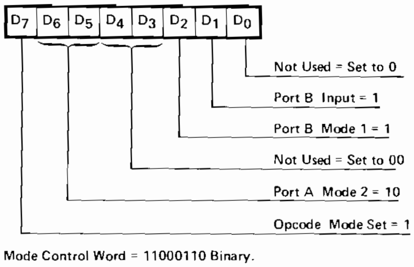
Figure 9. Mode 2 Status Word

Example #5 – MODE 2 STATUS WORD

If the 8255 is to be configured as follows:

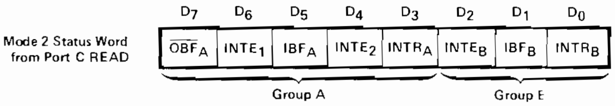
Port A	Mode 2 Bidirectional Bus
Port B	Mode 1 Input

The following mode control word is used:



Mode Control Word = 11000110 Binary.

After the 8255 mode control word has been issued, a read of Port C will obtain the following Mode 2 status word:


Mode 2 Status Word from Port C READ
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
$\overline{OBF}_A$ | $INTE_1$ | IBF_A | $INTE_2$ | $INTR_A$ | $INTE_B$ | IBF_B | $INTR_B$
Group A
Group E

Example #6 – MODE 2 INTERRUPT ENABLE/ DISABLE

The Mode 2 status word shown in Figure 9 contains two interrupt enable bits:

$INTE_1$ – Bit 6 – Enable output interrupts
$INTE_2$ – Bit 4 – Enable input interrupts

Bit set/reset control words may be constructed which may be used to control the INTE bits.

Set Bit 6 (Enable Output Interrupts) =
00001101 Binary

Reset Bit 6 (Disable Output Interrupts) =
00001100 Binary

Set Bit 4 (Enable Input Interrupts) =
00001001 Binary

Reset Bit 4 (Disable Input Interrupts) =
00001000 Binary

The control words shown were constructed from the standard bit set/reset format shown in Figure 7.

The value of CWR used in the following program example corresponds to the 8080 configuration shown in Figure 3.

```
CWR     EQU     0FBH            ; 8255 =1 CONTROL WORD REGISTER
; *****
        ENABLE INTERRUPTS FOR MODE 2 OUTPUT (SET PORT C BIT 6)
; *****
        MVI     A. 00001101B    ; GET SET BIT 6 CONTROL WORD
        OUT     CWR             ; OUTPUT TO 8255 =1 CONTROL WORD REGISTER
; *****
;       DISABLE INTERRUPTS FOR MODE 2 OUTPUT (RESET PORT C BIT 6)
; *****
        MVI     A, 00001100B    ; GET RESET BIT 6 CONTROL WORD
        OUT     CWR             ; OUTPUT TO 8255 =1 CONTROL WORD REGIS ˉER
```

SOFTWARE CONSIDERATIONS

Regardless of the mode selected, the software must always issue the correct mode control word after a reset of the device. Generally, an initialization routine is constructed which issues the correct mode control word, sets up the initial state of the control lines, and initializes any program internal data.

Many of the software requirements of the 8255 vary as a function of the mode selected. The simplest mode supported by the device is Mode 0 (Basic Input/Output). Generally, Mode 0 is used for simple status driven device interfaces (no interrupts). Figure 10 illustrates sample software that could be used to support such interfaces. Most devices support a BUSY or READY signal which is used to determine when the device is ready to input or output data and a DATA STROBE which is used to request data transfer (DATA STROBE may easily be generated with the Port C bit set/reset feature). In the Mode 0 configuration, Ports A and B are used to input/output byte oriented data. Port C is used to input 8255 status, peripheral status and to drive peripheral control lines.

When the Mode 1 and Mode 2 configurations are used the software is generally required to support interrupts. Software routines written for an interrupt driven environment tend to be more complex than status driven routines. The added complexity is due to the fact that interrupt driven systems are constructed such that other software tasks are run while the I/O transaction is in progress. A software routine that controls a peripheral device is generally referred to as a device driver. One method of implementing an interrupt driven device driver is to partition the device driver into a "Command Processor" and an "Interrupt Service Routine". The command processor is the module that validates and initiates user program requests to the device driver. A common method of passing information between the various software programs is to have the requesting routine provide a device control block in memory. A sample device control block is shown in Table II.

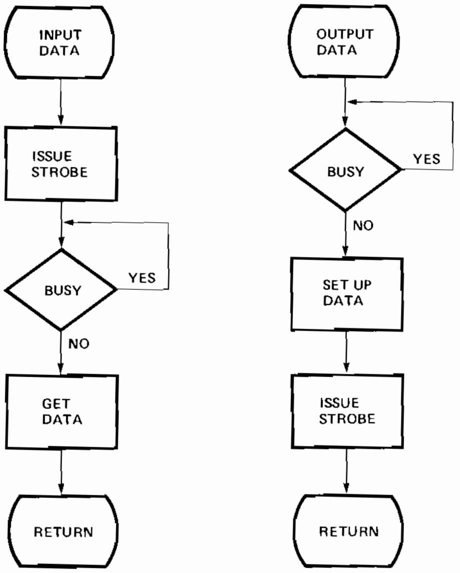


Figure 10. Sample Status Driven Software Flowchart

Table II. Sample Device Control Block

NAME	DESCRIPTION
Status	This 1-byte field is used to transmit the status of the I/O transaction (busy, complete, etc.).
Opcode	This 1-byte field defines the type of I/O (READ, WRITE, etc.).
Buffer Address	This 2-byte field specifies the source/destination of the data block.
Character Count	This 1-byte field is a count of the number of characters involved in the transaction.
Character Transferred Count	This 1-byte count of the number of characters which were actually transferred.
Completion Address	This 2-byte field is the address of the user supplied completion routine which will be called after the I/O has been performed.

The command processor validates the transaction and initiates the operation described by the control block. Control is then returned to the requestor so that other processing may proceed. The interrupt service routine processes the remainder of the transaction.

The interrupt service routine supports the following functions:

1. The state of the machine (registers, status, etc.) must be saved so that it may be restored after the interrupt is processed.
2. The source of the interrupt must be determined. The hardware may support a register which indicates the interrupting device, or the software may poll the devices through interrogating the Port C status word of each 8255.
3. Data must be passed to or from the device.
4. Control must be passed to the requesting routine at the completion of the I/O.
5. The state of the machine must be restored before returning to the interrupted program.

Figures 11 and 12 are simplified flowcharts of one of the many methods of implementing command processor and interrupt service routine modules.

The rest of this application note presents specific application examples. All of the 8080 assembly language programs supplied with the application examples use the standard Intel 8080 assembly language mnemonics. The programs discussed use the program equate statement to specify all hardware related data. Equate statements are used so that all references to an I/O port may be changed through a simple reassignment of the port address in the equate statement.

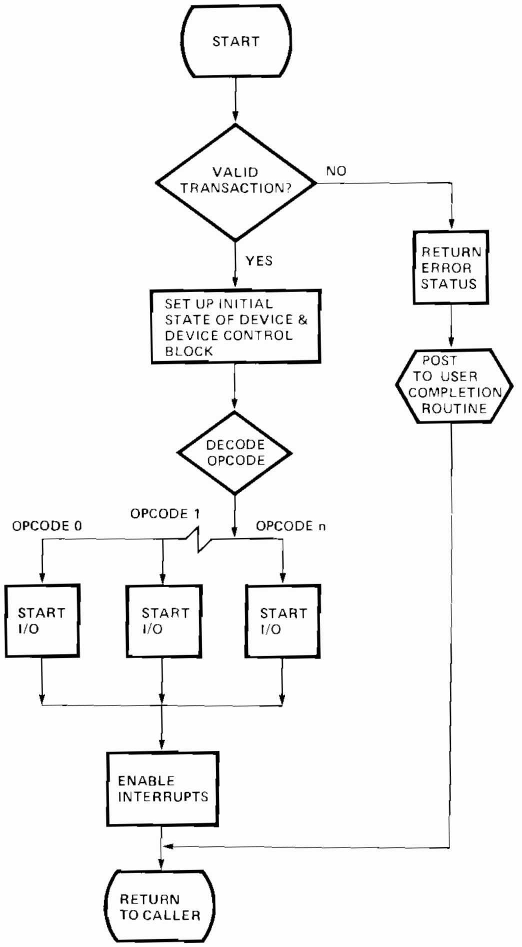


Figure 11. Command Processor

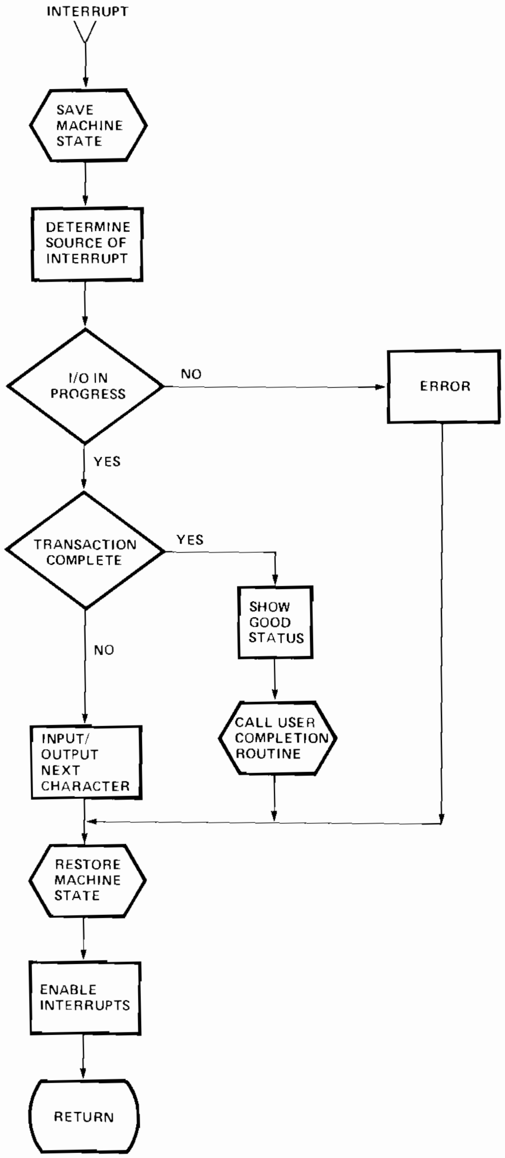


Figure 12. Interrupt Service Routine

MODE 0 – STATUS DRIVEN PERIPHERAL INTERFACE

This design example shows how a single 8255 in Mode 0 may be used to develop a status driven interface (no interrupts) for the Centronics 306 character printer, the Remex paper tape punch, and the Remex paper tape reader.

8255 To Peripheral Hardware Interface

The first step in the design is to examine the specification for the peripheral devices and identify the control and data signals which must be supported by the interface. Table III lists the signals which were chosen to be supported by the 8255 interface. All three of the devices support the standard BUSY/DATA STROBE interface discussed previously (see Figure 10). Figure 13 is a block diagram of the interface design. The 8255 Port A is configured as a Mode 0 output port which is used to support the printer and the paper tape punch data bus. Port B is configured as a Mode 0 input port and is used to input the paper tape reader data. Three of the Port C lower bits (PC_2–PC_0) configured in input mode are used to input the device busy indications. Three of the Port C upper bits (PC_6–PC_4) configured in output mode are used to support the device strobe signals required by each device.

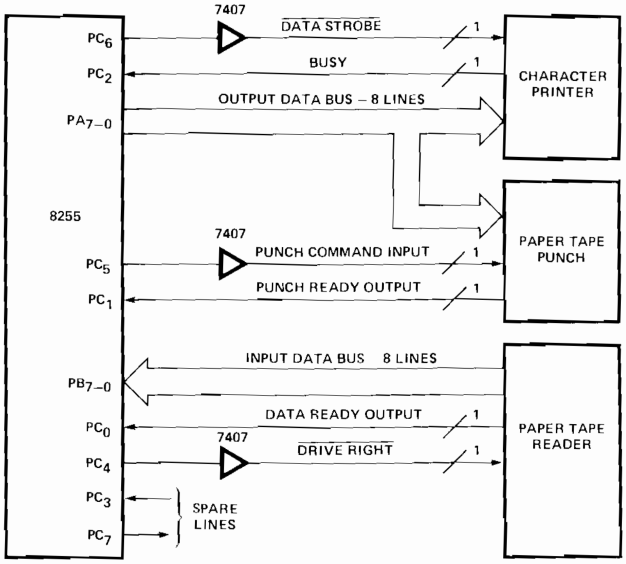
Table III. Peripheral Interface Signals

CHARACTER PRINTER	
Name:	DATA 0–DATA 7
Definition:	Input data levels. A high signal represents a binary 1 and a low signal represents a binary 0. These eight lines are the data lines to the printer.
Name:	$\overline{\text{DATA STROBE}}$
Definition:	A 0.5 μsec pulse (minimum) used to transfer data from the 8255 to the printer.
Name:	BUSY
Definition:	The level indicating that the printer cannot receive data.
PAPER TAPE PUNCH	
Name:	TRACKS 1–8 DATA INPUT
Definition:	Input data levels. A high signal causes a hole to be punched on the associated track. These eight lines are the data lines to the printer.
Name:	PUNCH COMMAND INPUT
Definition:	A true condition moves the tape and initiates punching the tape. This signal is actually a data strobe.
Name:	PUNCH READY OUTPUT
Definition:	True signal indicates that the punch is ready to accept a punch command. This is the punch busy line.
PAPER TAPE READER	
Name:	DATA TRACK OUTPUTS
Definition:	True signal indicates data track hole. These eight lines are the data lines from the punch.
Name:	$\overline{\text{DRIVE RIGHT}}$
Definition:	True signal drives the tape to the right and reads a character. This signal is actually the data strobe (initiate read signal).
Name:	DATA READY OUTPUT
Definition:	True signal indicates data track outputs are in "On character" condition. This signal is the reader busy line.

The drive requirements of the interface lines are a function of the peripheral interface circuitry, the length of the interface cable, and the environment in which the unit is running. In this particular design example, all output lines from the 8255 to the peripherals were buffered through a 7407 buffer/driver. The input lines from the peripherals were fed directly into the Port C and Port B inputs.

8080 CPU Module To 8255 Interface

The schematic of the completed hardware design is shown in Figure 14. The CPU module design shown is the design which was implemented for Intel's SDK 80 kit board. The 8255 is addressed through the use of an isolated I/O architecture utilizing a linear select scheme. Address bits A_1 and A_0 are used to select the 8255 port. Address bit A_3 is the exclusive enable for 8225 #1. Examination of the schematic shows that all of the 8255 interface lines are directly driven by the CPU module.



NOTE:
1. OUTPUT DATA BUS BUFFERED WITH 7407.
2. ALL 8255 OUTPUT LINES ARE PULLED UP TO +5V AT THE PERIPHERAL.

Figure 13. Interface Block Diagram

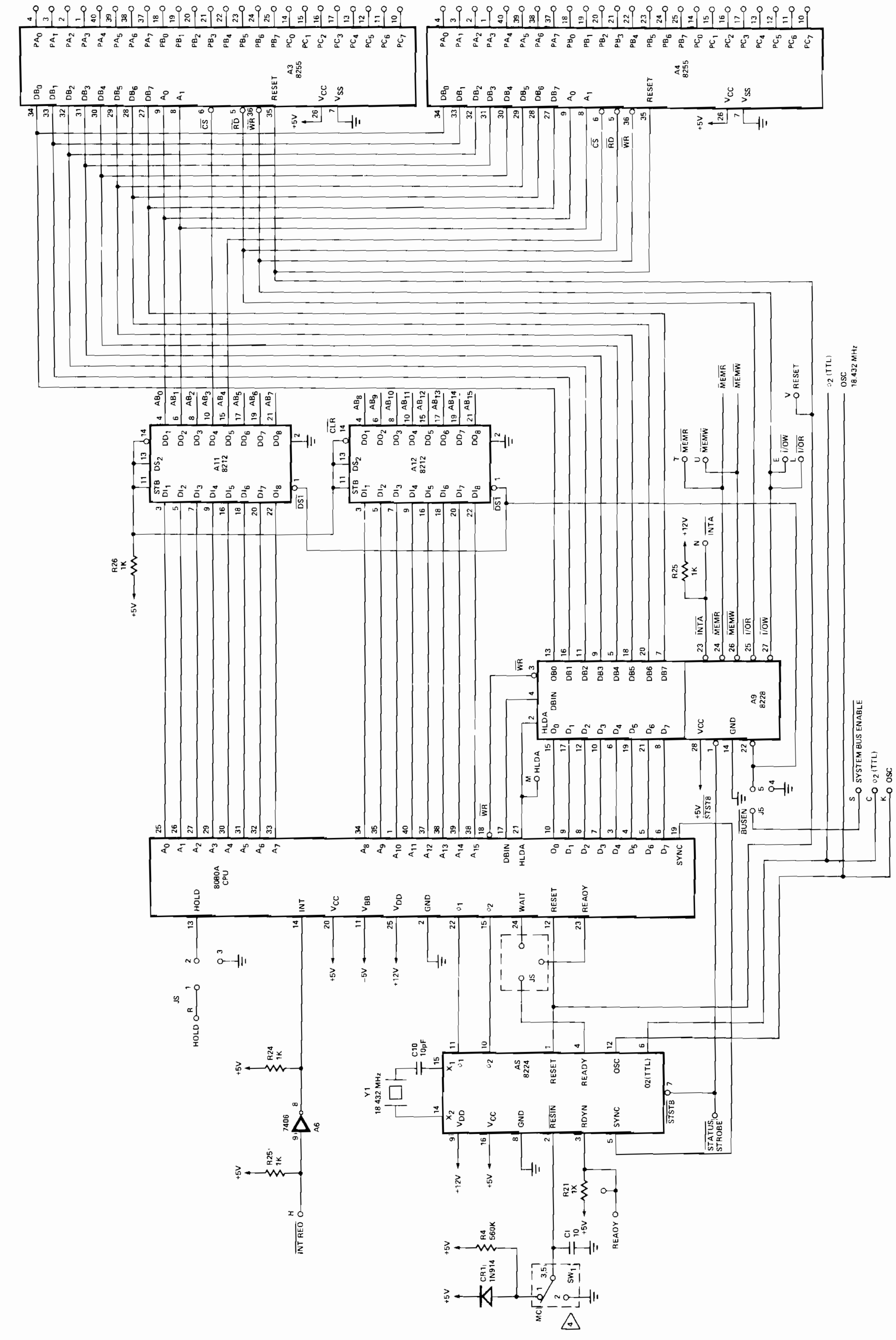
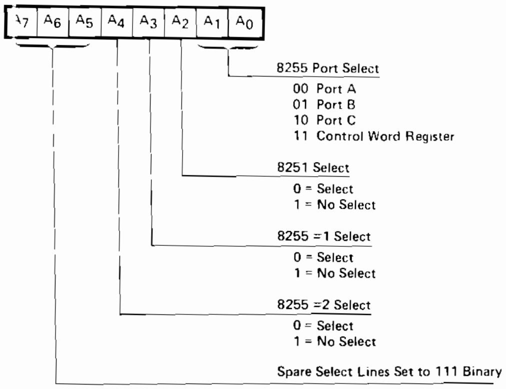


Figure 14. SDK 80 Schematic

Mode 0 Interface Software

An initialization routine and three device drivers (one for each peripheral device) are required to support the peripheral interface. The I/O port addresses implemented by the hardware are shown in Figure 15. The unused chip select bits are set to one so that chip select conflicts will not result if the unused bits are required by an expanded system.



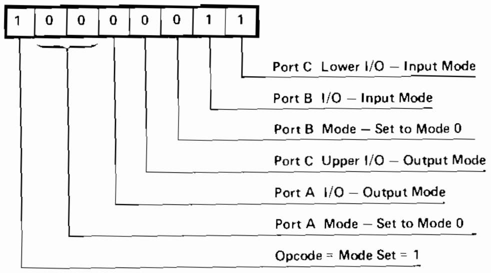
Port Selected	Port Select Character (In Hexadecimal)
Port A 8255 =1	F4
Port B 8255 =1	F5
Port C 8255 =1	F6
Control Word Register 8255 =1	F7
Port A 8255 =2	EC
Port B 8255 =2	ED
Port C 8255 =2	EE
Control Word Register 8255 =2	EF

Figure 15. I/O Port Addresses

Note that the initialization routine issues the mode control word (shown in Figure 16). It also sets the low true DATA STROBE signals to an inactive (high) state.

Hardware Requirements:

Port A – Output Mode 0
Port C Upper Bits C_7–C_4 – Output Mode 0
Port C Lower Bits C_3–C_0 – Input Mode 0
Port B – Input Mode 0



1 0 0 0 0 0 1 1
Port C Lower I/O – Input Mode
Port B I/O – Input Mode
Port B Mode – Set to Mode 0
Port C Upper I/O – Output Mode
Port A I/O – Output Mode
Port A Mode – Set to Mode 0
Opcode = Mode Set = 1

Mode Control Word = 100000011 Binary = 83 HEX.

Figure 16. Mode Control Word

```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 1
MODE ZERO EXAMPLE


                        TITLE 'MODE ZERO EXAMPLE'
                ;*****
                ;
                ;       CHARACTER PRINTER, PAPER TAPE PUNCH, PAPER TAPE READER
                ;       MODE ZERO EXAMPLE
                ;
                ;*****
                ;
                ;*****
                ;       PROGRAM EQUATES
                ;*****
  00F4          PORTA   EQU     0F4H    ; 8255 PORT A
                PORTB   EQU     0F5H    ; 8255 PORT B
  00F6          PORTC   EQU     0F6H    ; 8255 PORT C
  00F7          CWR     EQU     0F7H    ; 8255 CONTROL WORD REGISTER
                ;*****
                ;
                ;       INITIALIZATION CONTROL WORD
                ;
                ;               USED TO CONFIGURE THE 8255 AS FOLLOWS:
                ;
                ;                       PORT A - OUTPUT MODE ZERO
                ;                       PORT B - INPUT MODE ZERO
                ;                       PORT C (UPPER) - OUTPUT
                ;                       PORT C (LOWER) - INPUT
                ;
                ;*****
  0083          ICW     EQU     10000011B       ; INITIALIZATION CONTROL WORD
                ;*****
                ;
                ;       SET/RESET CONTROL WORDS FOR GENERATION OF DATA STROBES
                ;       ON PORT C.
                ;
                ;*****
  000D          LPSON   EQU     00001101B       ; PRINTER DATA STROBE ON
  000C          LPSOF   EQU     00001100B       ; PRINTER DATA STROBE OFF
  000B          PNSON   EQU     00001011B       ; PUNCH DATA STROBE ON
  000A          PNSOF   EQU     00001010B       ; PUNCH DATA STROBE OFF
  0009          RDSON   EQU     00001001B       ; READER DATA STROBE ON
  0008          RDSOF   EQU     00001000B       ; READER DATA STROBE OFF
                ;*****
                ;       BIT MASK FOR DEVICE BUSY CHECK
                ;*****
  0004          LPBSY   EQU     04H     ; LINE PRINTER BUSY
  0002          PNBSY   EQU     02H     ; PUNCH BUSY
  0001          RDBSY   EQU     01H     ; READER BUSY
```

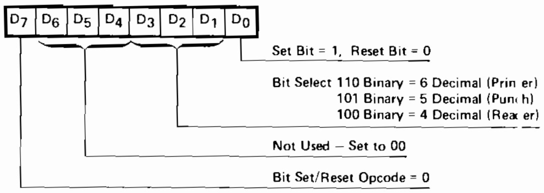
```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 2
MODE ZERO EXAMPLE - INITIALIZATION ROUTINE


                ;*****
                ;       PROGRAM ORIGIN
                ;*****
  3000                  ORG     03000H
                ;*****
                ;       INITIALIZATION ROUTINE
                ;       A REGISTER MODIFIED
                ;*****
                INIT:
  3000 3E83             MVI     A,ICW   ; GET INITIALIZATION CONTROL WORD
  3002 D3F7             OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
                ;*****
                ;       SET ALL LOW TRUE DATA STROBES ON
                ;*****
  3004 3E0D             MVI     A,LPSON ; GET CONTROL WORD TO TURN ON PRINTER DATA STROBE
  3006 D3F7             OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
  3008 3E09             MVI     A,RDSON ; GET CONTROL WORD TO TURN ON READER DATA STROBE
  300A D3F7             OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
  300C C9               RET             ; RETURN TO CALLER
```

The three peripheral drivers which follow all have the basic structure discussed previously. Consider the printer routine. Here the user routine places an ASCII data character in the C-register and passes control to the LPST location through a subroutine call. The printer driver interrogates the status of the printer by reading Port C. If the printer is busy, the routine will loop until the printer is idle. When the printer is ready to accept a data character, the character is placed on the Port A lines and a DATA STROBE is generated. After generating the DATA STROBE, the driver executes a subroutine return to the caller.

The DATA STROBE signals to the devices are generated through the use of the Port C bit set/reset feature. The bit set/reset control words used are shown in Figure 17.

Summary/Conclusions

This design example discussed the basic hardware and software required to handle a simple device interface. The 8255 will easily accommodate a more complex interface design which utilizes additional interface lines supported by the peripheral.

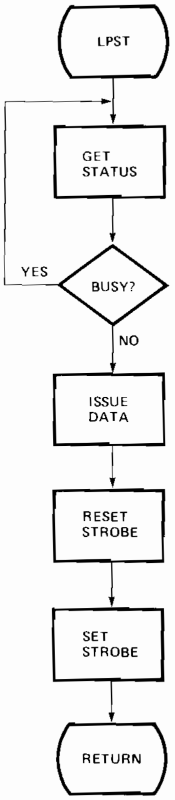

D7
D6
D5
D4
D3
D2
D1
D0
Set Bit = 1, Reset Bit = 0
Bit Select 110 Binary = 6 Decimal (Prin er)
101 Binary = 5 Decimal (Pun h)
100 Binary = 4 Decimal (Rea er)
Not Used – Set to 00
Bit Set/Reset Opcode = 0

The control word for set Printer DATA STROBE (PC 6) = 00001101 binary.
The control word for reset Printer DATA STROBE (PC 6) = 00001100 binary.
The control word for set Punch DATA STROBE (PC 5) = 00001011 binary.
The control word for reset Punch DATA STROBE (PC 5) = 00001010 binary.
The control word for set Reader DATA STROBE (PC 4) = 00001001 binary.
The control word for reset Reader DATA STROBE (PC 4) = 00001000 binary.

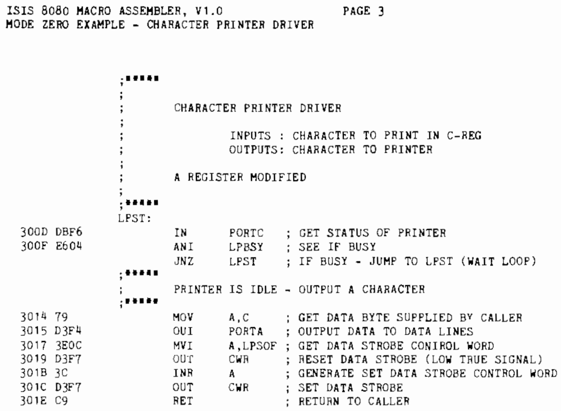
Figure 17. Bit Set/Reset Control Words

For instance, one of the spare Port C output lines may be used to control the punch direction Support of this additional feature would require minor modification of the device driver so that the punch direction line could be specified by the user routine.

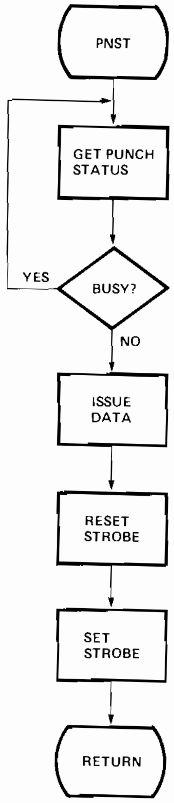
Through consideration of this example, the use of the 8255 in Mode 0 should become evident.



PRINTER DRIVER

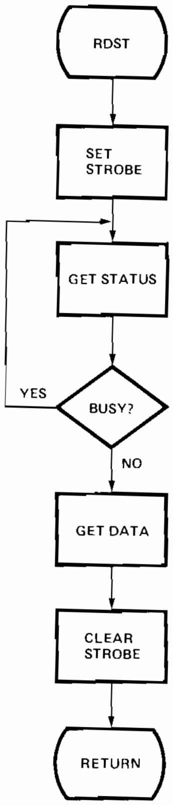


```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 3
MODE ZERO EXAMPLE - CHARACTER PRINTER DRIVER


                ;*****
                ;
                ;       CHARACTER PRINTER DRIVER
                ;
                ;               INPUTS : CHARACTER TO PRINT IN C-REG
                ;               OUTPUTS: CHARACTER TO PRINTER
                ;
                ;       A REGISTER MODIFIED
                ;
                ;*****
                LPST:
 300D DBF6              IN      PORTC   ; GET STATUS OF PRINTER
 300F E604              ANI     LPBSY   ; SEE IF BUSY
                        JNZ     LPST    ; IF BUSY - JUMP TO LPST (WAIT LOOP)
                ;*****
                ;       PRINTER IS IDLE - OUTPUT A CHARACTER
                ;*****
 3014 79                MOV     A,C     ; GET DATA BYTE SUPPLIED BY CALLER
 3015 D3F4              OUI     PORTA   ; OUTPUT DATA TO DATA LINES
 3017 3E0C              MVI     A,LPSOF ; GET DATA STROBE CONIROL WORD
 3019 D3F7              OUT     CWR     ; RESET DATA STROBE (LOW TRUE SIGNAL)
 301B 3C                INR     A       ; GENERATE SET DATA STROBE CONTROL WORD
 301C D3F7              OUT     CWR     ; SET DATA STROBE
 301E C9                RET             ; RETURN TO CALLER
```



PUNCH DRIVER

```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 4
MODE ZERO EXAMPLE - PAPER TAPE PUNCH DRIVER


                    ;*****
                    ;
                    ;       PAPER TAPE PUNCH DRIVER
                    ;
                    ;               INPUTS : DATA TO PUNCH IN C-REGISTER
                    ;               OUTPUTS: DATA TO PUNCH
                    ;
                    ;       A REGISTER MODIFIED
                    ;
                    ;*****
                    PNST:
 301F DBF6                  IN      PORTC   ; GET STATUS OF PUNCH
 3021 E602                  ANI     PNBSY   ; SEE IF BUSY
 3023 C21F30                JNZ     PNST    ; IF BUSY - JUMP TO PNST (WAIT LOOP)
                    ;*****
                    ;       PUNCH IS IDLE - OUTPUT A CHARACTER
                    ;*****
 3026 79                    MOV     A,C     ; GET DATA BYTE SUPPLIED BY CALLER
                            OUT     PORTA   ; OUTPUT DATA TO DATA LINES
 3029 3E0B                  MVI     A,PNSON ; SET DATA STROBE CONTROL WORD
 302B D3F7                  OUT     CWR     ; SET DATA STROBE
 302D 3D                    DCR     A       ; GENERATE RESET DATA STROBE CONTROL WORD
 302E D3F7                  OUT     CWR     ; RESET DATA STROBE
 3030 C9                    RET             ; RETURN TO CALLER
```



READER DRIVER

```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 5
MODE ZERO EXAMPLE - PAPER TAPE READER DRIVER


                    ;*****
                    ;
                    ;       PAPER TAPE READER DRIVER
                    ;
                    ;               INPUTS : DATA FROM READER
                    ;               OUTPUTS: CHARACTER TO USER IN C-REGISTER
                    ;
                    ;       A AND C REGISTER MODIFIED
                    ;
                    ;*****
                    RDST:
 3031 3E08                  MVI     A,RDSOF ; GET STROBE CONTROL WORD (LOW TRUE SIGNAL)
 3033 D3F7                  OUT     CWR     ; SET DATA STROBE
                    RDLP:
 3035 DBF6                  IN      PORTC   ; CEI STATUS OF DEVICE
 3037 E601                  ANI     RDBSY   ; SEE IF BUSY
 3039 C23530                JNZ     RDLP    ; IF BUSY - LOOP UNTIL IDLE
                    ;*****
                    ;       READER NOT BUSY - GET CHAR AND CLEAR STROBE
                    ;*****
 303C DBF5                  IN      PORTB   ; GET CHARACIER
 303E 0E07                  MVI     C,A     ; SAVE CHARACTER
 3040 3E09                  MVI     A,RDSON ; GET STROBE SET CONTROL WORD (LOW TRUE SIGNAL)
 3042 D3F7                  OUT     CWR     ; TURN OFF STROBE
 3044 C9                    RET             ; RETURN TO CALLER
                    ;*****
                    ;       END OF MODE ZERO EXAMPLE
                    ;*****
 0000                       END
```

MODE 1 INTERRUPT DRIVEN PRINTER INTERFACE

The status driven interface described in the previous example required the software driver to poll the device status for completion. An alternate approach is to construct the device interface such that an interrupt is used to signal the completion of the operation. When an interrupt driven interface is utilized, the time that was dedicated to polling can be used to perform other functions and the effective processor through-put is increased. This example demonstrates how an 8255 configured in Mode 1 may be used to develop an interrupt driven interface for the Centronics 306 character printer.

CPU Module To 8255 Interface

The 8080 bus interface implemented for this example is the same as the Mode 0 example with the addition of interrupt support. Interrupt support is implemented through the use of a special feature of the 8228 System Controller. If only one interrupt vector is required (such as in small systems), the 8228 can automatically assert an RST 7 instruction onto the data bus at the proper time. This option is selected by connecting the $\overline{\text{INTA}}$ output of the 8228 to the +12-volt supply through a 1K ohm series resistor.

The Mode 1 interrupt support logic of the 8255 provides an interrupt request line for each port. The 8255 interrupt request line (INTR_A) must be connected to the INT line of the 8080. A 10K ohm pullup resistor is used to insure that the V_{IH} requirements of the 8080 are met.

8255 To Peripheral Interface

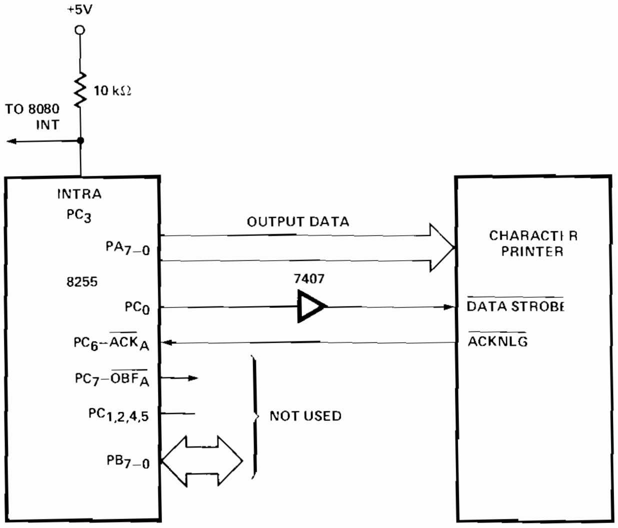
The interrupt driven configuration control signal interface to the printer is different than the status driven interface. Instead of a BUSY/DATA STROBE interface, a DATA STROBE/ACK interface is supported. The ACK signal notifies the 8255 that a character transferred to the printer by a DATA STROBE has been accepted. After an ACK is issued the printer is considered idle. The block diagram shown in Figure 18 displays the interface signals used.

The Mode 1 interrupt driven peripheral support signals used are:

PA_7–PA_0 – Output Data
Used to support the printer data port.

$\overline{\text{OBF}}$ – Output Buffer Full
This line goes low when data is placed in the output buffer. The OBF signal may be used as a data strobe signal when interfacing to peripherals which do not require a pulsed input. The Centronics 306 requires a pulsed DATA STROBE signal. This signal is supported by Port C bit 0.

$\overline{\text{ACK}}$ – ACKnowledge
This line is used to signal the 8255 that the device has accepted the data. This line is supported by the printer ACKNLG signal.



NOTES:
1. DATA BUS BUFFERED WITH 7407.
2. ALL 8255 OUTPUT LINES ARE PULLED UP TO +5V AT THE PERIPHERAL.

Figure 18. Interface Block Diagram

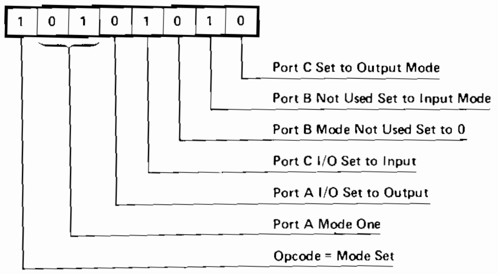
Mode 1 Software Driver

The software driver implemented for this example utilizes the typical interrupt driven software structure outlined previously. The initialization routine issues the mode control word (shown in Figure 19) to the 8255 after reset of the device. The initialization routine also places a jump to the interrupt service routine in the interrupt location for RST 7. The command processor is started by the user routine through a subroutine call to PSTRT, with the address of the control block in the D and E registers (the control block format is shown in Table IV). The command processor insures that an I/O operation is not already in progress, starts the I/O, enables interrupts, and returns to the caller so that other processing may proceed.

After a character is placed in the output buffer, the DATA STROBE signal is generated through the use of the Port C bit set/reset feature. When the ACK is generated by the printer, the buffer full indication is cleared and the 8255 generates an interrupt. If interrupts are enabled, the interrupt request is serviced by the 8080 CPU through disabling processor interrupts and then executing the instruction at location 38 hexadecimal in program memory. The interrupt service routine saves the processor state and polls the 8255 to determine the source of the interrupt. Once the interrupting device is located, the control block is used to locate the next data character for transfer to the 8255 output buffer. After the entire buffer has been printed, the interrupt service routine passes control to the user-supplied completion routine. Before returning from the interrupt, the state of the processor is restored.

Hardware Requirements:

Port A – Input Mode 1
Port C (Lower) – Output
Port B
Port C (Upper) } Not Used


1
0
1
0
1
0
1
0
Port C Set to Output Mode
Port B Not Used Set to Input Mode
Port B Mode Not Used Set to 0
Port C I/O Set to Input
Port A I/O Set to Output
Port A Mode One
Opcode = Mode Set

Mode Control Word = 10101010 Binary = AA HEX.

Figure 19. Mode Control Word

Table IV. Printer Software Control Block

NAME	POSITION	DEFINITION
Status	Byte 0	A 1-byte field which defines the completion status of an I/O. 00 = Good completion 01 = Error – command already in progress
Buffer Address	Byte 1, 2	Pointer to the start of the data to print.
Character Count	Byte 3	Count of the number of characters to print.
Character Transferred Count	Byte 4	The number of characters transferred.
Completion Address	Byte 5, 6	Address of a user supplied routine which will be called after the I/O has been performed.

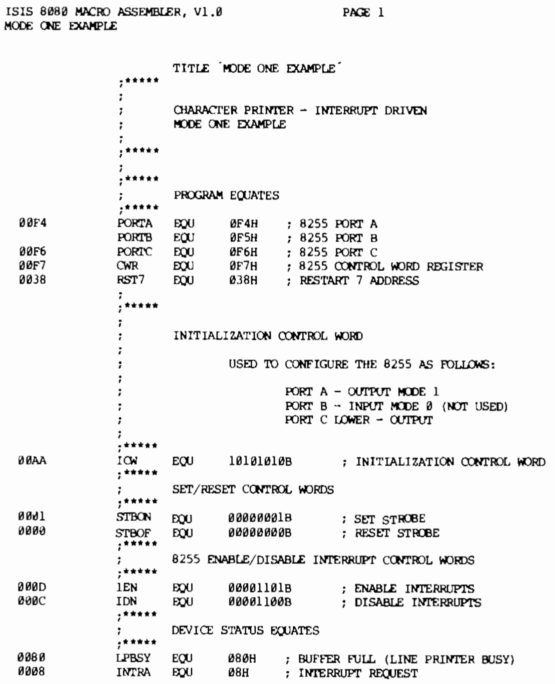
NOTES:

1. An opcode field is not required because WRITE is the only operation performed.
2. The control block must reside above location FF Hex.

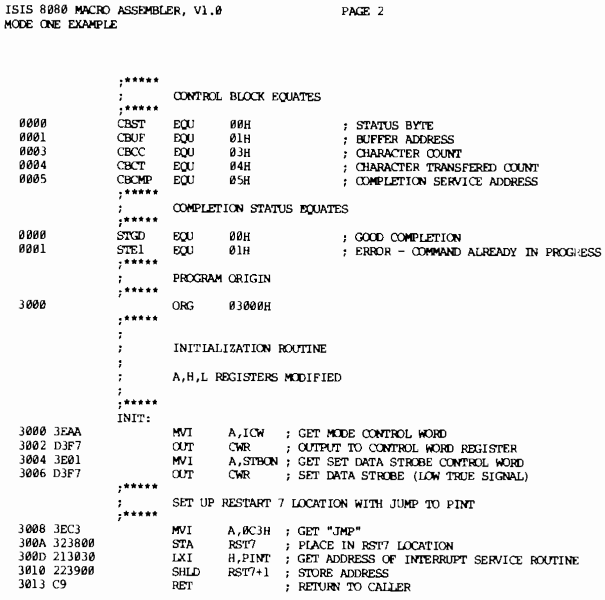
There are a number of error conditions which may occur, such as an interrupt from a device which does not have a control block in progress, or an interrupt when polling establishes that no device requires service. Neither of these errors should occur, but if they do, the driver should perform in a consistent fashion. The recovery routines implemented to handle error conditions are determined by the particular applications environment.

Summary/Conclusions

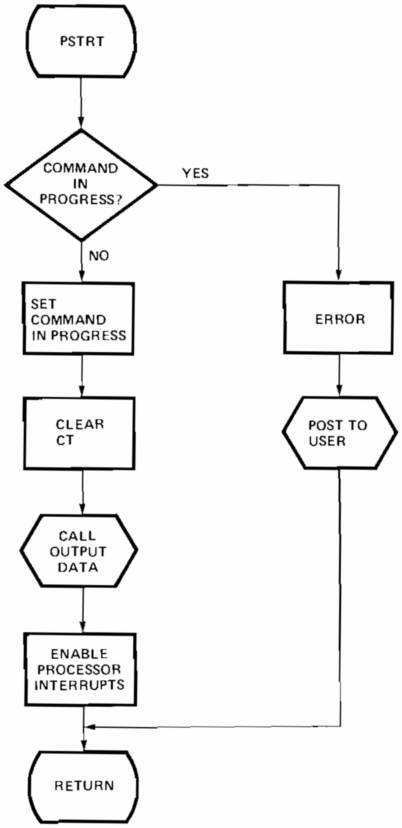
When utilized in a small system design, the 8255 interrupt support logic provides all of the capabilities required to implement an interrupt driven hardware interface without the use of external logic. In larger system designs, the designer may chose to use additional hardware to determine the source of interrupt requests without software polling. The software design required by an interrupt driven system is inherently more complex than the status driven interface. If an interrupt driven system is required the added complexity is a small price to pay for a significant increase in system through-put.



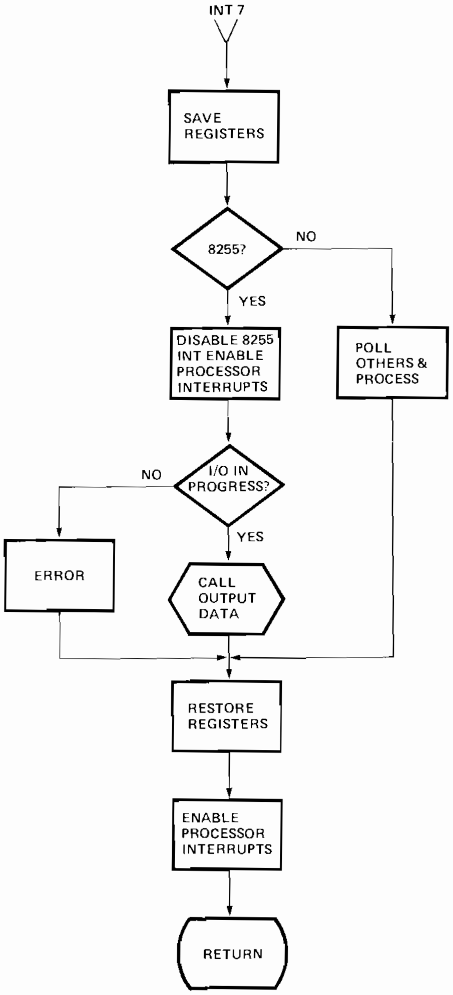
```
ISIS 8Ø8Ø MACRO ASSEMBLER, V1.Ø                PAGE 1
MODE ONE EXAMPLE

                        TITLE 'MODE ONE EXAMPLE'
               ;*****
               ;
               ;        CHARACTER PRINTER - INTERRUPT DRIVEN
               ;        MODE ONE EXAMPLE
               ;
               ;*****
               ;
               ;*****
               ;        PROGRAM EQUATES
               ;*****
 ØØF4          PORTA    EQU     ØF4H    ; 8255 PORT A
               PORTB    EQU     ØF5H    ; 8255 PORT B
 ØØF6          PORTC    EQU     ØF6H    ; 8255 PORT C
 ØØF7          CWR      EQU     ØF7H    ; 8255 CONTROL WORD REGISTER
 ØØ38          RST7     EQU     Ø38H    ; RESTART 7 ADDRESS
               ;
               ;*****
               ;
               ;        INITIALIZATION CONTROL WORD
               ;
               ;                USED TO CONFIGURE THE 8255 AS FOLLOWS:
               ;
               ;                        PORT A - OUTPUT MODE 1
               ;                        PORT B - INPUT MODE Ø (NOT USED)
               ;                        PORT C LOWER - OUTPUT
               ;
               ;*****
 ØØAA          ICW      EQU     1Ø1Ø1Ø1ØB       ; INITIALIZATION CONTROL WORD
               ;*****
               ;        SET/RESET CONTROL WORDS
               ;*****
 ØØØ1          STBON    EQU     ØØØØØØØ1B       ; SET STROBE
 ØØØØ          STBOF    EQU     ØØØØØØØØB       ; RESET STROBE
               ;*****
               ;        8255 ENABLE/DISABLE INTERRUPT CONTROL WORDS
               ;*****
 ØØØD          lEN      EQU     ØØØØ11Ø1B       ; ENABLE INTERRUPTS
 ØØØC          IDN      EQU     ØØØØ11ØØB       ; DISABLE INTERRUPTS
               ;*****
               ;        DEVICE STATUS EQUATES
               ;*****
 ØØ8Ø          LPBSY    EQU     Ø8ØH    ; BUFFER FULL (LINE PRINTER BUSY)
 ØØØ8          INTRA    EQU     Ø8H     ; INTERRUPT REQUEST
```



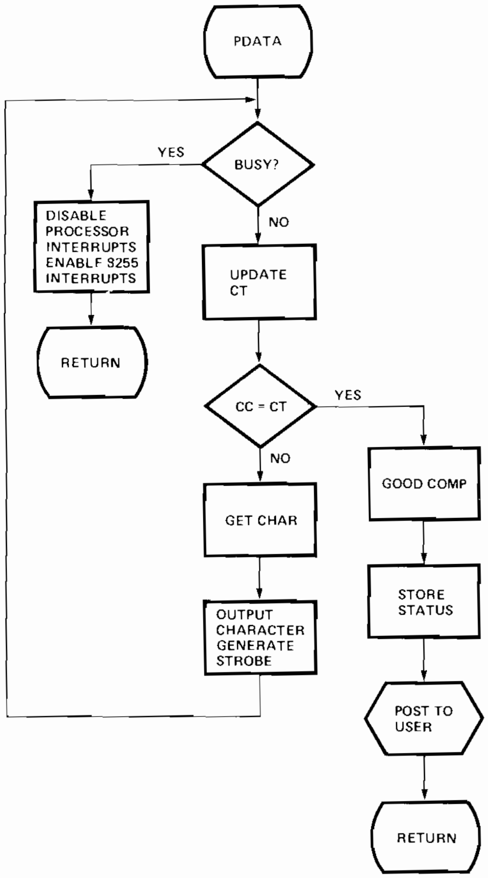
```
ISIS 8Ø8Ø MACRO ASSEMBLER, V1.Ø                PAGE 2
MODE ONE EXAMPLE

               ;*****
               ;        CONTROL BLOCK EQUATES
               ;*****
 ØØØØ          CBST     EQU     ØØH             ; STATUS BYTE
 ØØØ1          CBUF     EQU     Ø1H             ; BUFFER ADDRESS
 ØØØ3          CBCC     EQU     Ø3H             ; CHARACTER COUNT
 ØØØ4          CBCT     EQU     Ø4H             ; CHARACTER TRANSFERED COUNT
 ØØØ5          CBCMP    EQU     Ø5H             ; COMPLETION SERVICE ADDRESS
               ;*****
               ;        COMPLETION STATUS EQUATES
               ;*****
 ØØØØ          STGD     EQU     ØØH             ; GOOD COMPLETION
 ØØØ1          STE1     EQU     Ø1H             ; ERROR - COMMAND ALREADY IN PROGRESS
               ;*****
               ;        PROGRAM ORIGIN
               ;*****
 3ØØØ                   ORG     Ø3ØØØH
               ;*****
               ;
               ;        INITIALIZATION ROUTINE
               ;
               ;        A,H,L REGISTERS MODIFIED
               ;
               ;*****
               INIT:
 3ØØØ 3EAA              MVI     A,ICW   ; GET MODE CONTROL WORD
 3ØØ2 D3F7              OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
 3ØØ4 3EØ1              MVI     A,STBON ; GET SET DATA STROBE CONTROL WORD
 3ØØ6 D3F7              OUT     CWR     ; SET DATA STROBE (LOW TRUE SIGNAL)
               ;*****
               ;        SET UP RESTART 7 LOCATION WITH JUMP TO PINT
               ;*****
 3ØØ8 3EC3              MVI     A,ØC3H  ; GET "JMP"
 3ØØA 3238ØØ            STA     RST7    ; PLACE IN RST7 LOCATION
 3ØØD 21303Ø            LXI     H,PINT  ; GET ADDRESS OF INTERRUPT SERVICE ROUTINE
 3Ø1Ø 2239ØØ            SHLD    RST7+1  ; STORE ADDRESS
 3Ø13 C9                RET             ; RETURN TO CALLER
```



```
ISIS 8080 MACRO ASSEMBLER, V1.0                   PAGE 3
COMMAND PROCESSOR


                    ;*****
                    ;
                    ;       COMMAND PROCESSOR
                    ;
                    ;               INPUTS: CONTROL BLOCK ADDRESS IN D AND E REGISTERS
                    ;
                    ;               OUTPUTS: START I/O OR ERROR STATUS IN CONTROL BLOCK
                    ;
                    ;       A,H,L REGISTERS MODIFIED
                    ;
                    ;*****
                    PSTRT:
  3014 3AA230               LDA     PIPRG+1 ; GET PRINT IN PROGRESS BLOCK ADDRESS
  3017 A7                   ANA     A       ; SEE IF ZERO (PRINT IN PROGRESS)
                                            ; IF BLOCK ADDRESS NOT EQUAL TO ZERO THEN
                                            ; PRINT IN PROGRESS
  3018 C22B30               JNZ     PSTE    ; IF YES - BRANCH TO ERROR
  301B EB                   XCHG
  301C 22A130               SHLD    PIPRG   ; SAVE CONTROL BLOCK ADDRESS
  301F EB                   XCHG
  3020 210400               LXI     H,CBCT  ; GET INDEX TO CT
  3023 19                   DAD     D       ; COMPUTE ADDRESS OF CT
  3024 3600                 MVI     M,00H   ; CLEAR CT
  3026 CD5830               CALL    PDATA   ; START I/O
  3029 FB                   EI              ; ENABLE PROCESSOR INTERRUPTS
  302A C9                   RET             ; RETURN TO CALLER
                    ;*****
                    ;       ERROR - TRANSACTION ALREADY IN PROGRESS
                    ;*****
                    PSTE:
  302B 3E01                 MVI     A,STE1  ; GET ERROR STATUS CODE
  302D C39430               JMP     POST    ; PASS CONTROL TO COMPLETION ROUTINE
```



```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 4
PRINTER INTERRUPT SERVICE ROUTINE


                    ;*****
                    ;       PRINTER INTERRUPT SERVICE ROUTINE
                    ;       ALL REGISTERS SAVED AND RESTORED
                    ;*****
                    PINT:
  3030 F5                   PUSH    PSW     ; SAVE PSW
  3031 C5                   PUSH    B       ; SAVE REGISTER PAIR B AND C
  3032 D5                   PUSH    D       ; SAVE REGISTER PAIR D AND E
                            PUSH    H       ; SAVE REGISTER PAIR H AND L
                    ;*****
                    ;       POLL INTERRUPT SOURCE - SEE IF 8255
                    ;*****
  3034 DBF6                 IN      PORTC   ; GET STATUS OF DEVICE
  3036 E608                 ANI     INTRA   ; SEE IF INT
  3038 CA5230               JZ      PPOLL   ; NO - BRANCH TO POLL OTHER DEVICES IF ANY
  303B 3E0C                 MVI     A,IDN   ; GET 8255 INT DISABLE CONTROL WORD
  303D D3F7                 OUT     CWR     ; DISABLE DEVICE INTERRUPTS
  303F FB                   EI              ; ENABLE PROCESSOR INTERRUPTS
  3040 2AA130               LHLD    PIPRG   ; GET CONTROL BLOCK ADDRESS
  3043 AF                   XRA     A       ; CLEAR A REG
  3044 BC                   CMP     H       ; SEE IF PRINT IN PROGRESS
  3045 CA5530               JZ      PIER1   ; NO - BRANCH TO ERROR ROUTINE
  3048 EB                   XCHG
  3049 CD5830               CALL    PDATA   ; PRINT DATA
                    ;*****
                    ;       RESIORE REGISTERS AND RETURN FROM INTERRUPT
                    ;*****
                    PRIN:
  304C E1                   POP     H       ; RESTORE REGISTER PAIR H AND L
  304D D1                   POP     D       ; RESTORE REGISTER PAIR D AND E
  304E C1                   POP     B       ; RESTORE REGISTER PAIR B AND C
  304F F1                   POP     PSW     ; RESTORE PSW
  3050 FB                   EI              ; ENABLE PROCESSOR INTERRUPTS
  3051 C9                   RET             ; RETURN TO INTERRUPTED PROCESS
                    ;*****
                    ;       POLL OTHER DEVICES IF ANY
                    ;               IF NO OTHER DEVICES TO POLL - USER SUPPLIED ERROR
                    ;               RECOVERY ROUTINE.
                    ;*****
                    PPOLL:
  3052 C34C30               JMP     PRTN    ; RETURN
                    ;*****
                    ;       ERROR - INTERRUPT FROM IDLE DEVICE
                    ;               USER SUPPLIED ERROR RECOVERY ROUTINE
                    ;*****
                    PIER1:
  3055 C34C3U               JMP     PRTN    ; RETURN
```



```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 5
PRINTER OUTPUT DATA ROUTINE


                ;*****
                ;
                ;       PRINTER OUTPUT DATA ROUTINE
                ;
                ;       CONTROL BLOCK ADDRESS IN D AND E REG
                ;
                ;*****
                PDATA:
  3058 DBF6             IN      PORTC   ; GET STATUS OF DEVICE
  305A E680             ANI     LPBSY   ; SEE IF BUSY (BUFFER FULL)
  305C CA8430           JZ      PD10    ; IF BUSY - BRANCH
  305F 210400           LXI     H,CBCT  ; GET INDEX TO CT
  3062 19               DAD     D       ; COMPUTE ADDRESS OF CT
  3063 7E               MOV     A,M     ; GET CT
  3064 34               INR     M       ; INC CT
  3065 2B               DCX     H       ; DEC TO CC
  3066 BE               CMP     M       ; SEE IF EQUAL
  3067 CA8A30           JZ      PCOMP   ; IF EQUAL - DONE GO TELL USER
  306A 210100           LXI     H,CBUF  ; GET INDEX TO BUFFER ADDRESS
  306D 19               DAD     D       ; COMPUTE ADDRESS OF BUFFER ADDRESS
  306E D5               PUSH    D       ; SAVE D AND E REGISTERS
  306F 5E               MOV     E,M     ; GET LSB OF BUFFER ADDRESS
  3070 23               INX     H       ; INC TO NEXT BYTE
  3071 56               MOV     D,M     ; GET BUFFER MSB
  3072 2600             MVI     H,OOH   ; CLEAR H REG
  3074 6F               MOV     L,A     ; GET CT
  3075 19               DAD     D       ; COMPUTE CHARACTER ADDRESS
  3076 7E               MOV     A,M     ; GET CHARACTER
  3077 D3F4             OUT     PORTA   ; OUTPUT CHARACTER TO PRINTER
  3079 3E00             MVI     A,STBOF ; RESET DATA STROBE (LOW TRUE SIGNAL)
  307B D3F7             OUT     CWR
  307D 3C               TNR     A       ; GENERATE SET CONTROL WORD
  307E D3F7             OUT     CWR     ; SET DATA STROBE
  3080 D1               POP     D       ; RESTORE CONTROL BLOCK ADDRESS
  3081 C35830           JMP     PDATA   ; LOOP UNTIL BUSY


                ;*****
                ;       PRINTER BUSY - RETURN
                ;*****
                PD10:
  3084 F3               DI              ; DISABLE INTERRUPTS
  3085 3E0D             MVI     A,IEN   ; ENABLE DEVICE INTERRUPTS
  3087 D3F7             OUT     CWR     ; SET INTERRUPT ENABLE
  3089 C9               RET             ; RETURN TO CALLER
                ;*****
                ;       POST GOOD COMPLETION TO USER
                ;*****
                PCOMP:
  308A 3E00             MVI     A,STGD  ; GET GOOD STATUS CODE
                        CALL    POST    ; POST TO USER
  308F AF               XRA     A       ; CLEAR A REG
  3090 32A230           STA     PIPRG+1 ; CLEAR COMMAND IN PROGRESS ADDRESS
  3093 C9               RET             ; RETURN TO CALLER


                ;*****
                ;
                ;       POST TO USER COMPLETION ROUTINE
                ;
                ;               INPUTS : STATUS CODE IN A REG
                ;                        CONTROL BLOCK ADDRESS IN D AND E REG
                ;               OUTPUTS: PASSES CONTROL TO USER COMPLETION ADDRES;
                ;                        SPECIFIED IN CONTROL BLOCK
                ;                        WITH CONTROL BLOCK ADDRESS IN D AND E
                ;
                ;               A,H,L,B,C REG MODIFIED
                ;
                ;*****
                POST:
  3094 EB               XCHG
  3095 77               MOV     M,A     ; UPDATE STATUS
  3096 EB               XCHG
  3097 210500           LXI     H,CBCMP ; GET INDEX TO COMPLETION ADDRESS
  309A 19               DAD     D       ; COMPUTE ADDRESS
  309B 4E               MOV     C,M     ; GET LSB OF COMPLETION ADDRESS
  309C 23               INX     H       ; INC TO NEXT BYTE
  309D 46               MOV     B,M     ; GET MSB  OF COMPLETION ADDRESS
  309E C5               PUSH    B       ; PUSH ADDRESS INTO STACK
  309F C9               RET             ; PASS CONTROL TO USER ROUTINE
  30A0 C9               RET             ; RETURN TO CALLER
                ;*****
                ;       DATA AND TABLES
                ;*****
  30A1 0000     PIPRG:  DW      0       ; IN PROGRESS CONTROL BLGCK ADDRESS
                                        ; IF DATA = 0 NO CONTROL BLOCK IN PROGRESS
                                        ; IF DATA NOT EQUAL TO ZERO CONTROL BLOCK  N PROGRESS
                ;*****
                ;       END OF MODE ONE EXAMPLE
                ;*****
  0000                  END
```

MODE 2 – 8080 TO 8080 INTERFACE

Due to the drastic reduction of hardware costs, system designs which utilize multiple CPU Modules are becoming more common. An 8080 may be configured as a master CPU and used to control multiple 8080 slave modules which act as intelligent I/O controllers. When multiple CPUs are utilized, a method' of processor intercommunication must be supported. Figure 20 is a block diagram of one method of implementing a master/slave interface through the use of the 8255 Mode 2 bidirectional bus.

Hardware Discussion

Two complete 8080 systems are required for this example. Intel's SBC 80/10 OEM board is used as the master CPU module and Intel's SDK 80 board is used as the slave CPU. The SBC 80/10 supports an 8255 which is configured in Mode 2. The 8255 is selected through the use of a decoded select scheme. Through the use of the 8228 RST 7 interrupt feature, a simple interrupt structure is supported. The SDK 80 is configured without interrupts for this example. The external logic required for this example is associated with the slave CPU. Simple logic is implemented which allows the slave CPU to generate the $\overline{\text{ACK}}$ and $\overline{\text{STB}}$ signals required to READ from and WRITE to the 8255 bidirectional bus with a single I/O instruction.

The system shown in Figure 20 utilizes SSI logic to read the 8255 IBF and $\overline{\text{OBF}}$ signals. If two spare 8255 input lines are available they could be used to input the IBF and $\overline{\text{OBF}}$ signals and eliminate the SSI logic.

Software Discussion

Two sets of software are required to support the processor to processor interface. The master resident software which follows conforms to the simple interrupt driven software structure outlined previously. The initialization routine issues the Mode 2 control word to the 8255 after device reset. The command processor accepts READ/WRITE control blocks which provide a simple user interface for transferring data to/from the slave CPU. The master software is capable of processing both a read and a write control block simultaneously. The slave resident software shown at the end of this example utilizes the status driven approach.

Summary/Conclusions

It is important to note that this design may be expanded to include more slave CPUs by simply adding another 8255 to the master module for each slave. The software drivers discussed address only the passing of data between the two processors. Specific applications generally dictate a software protocol be implemented for information transfer.

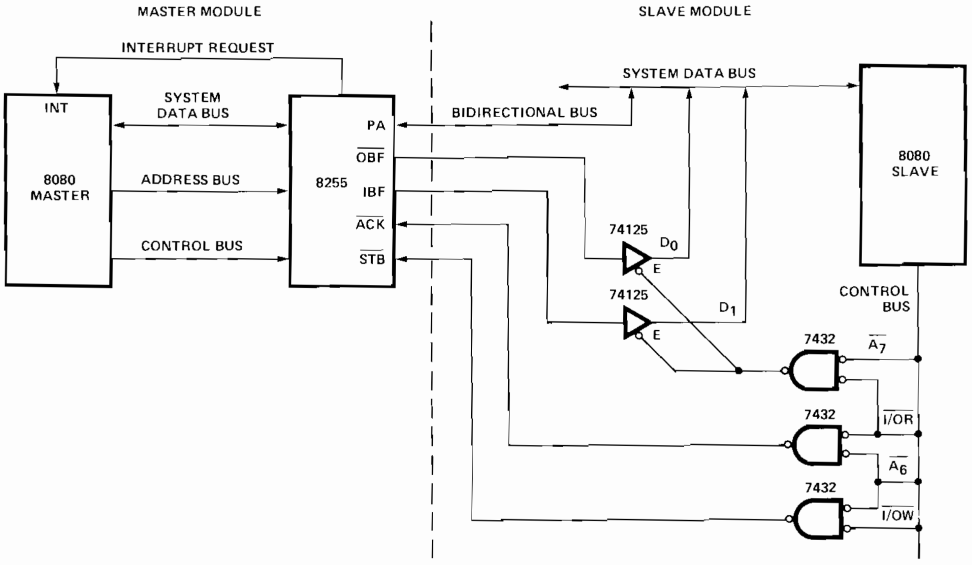
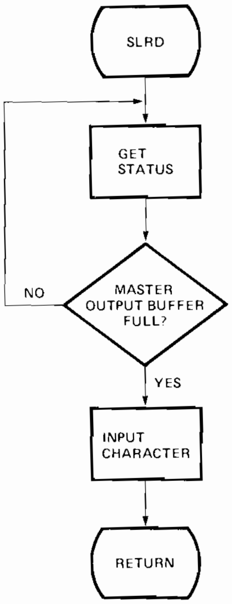
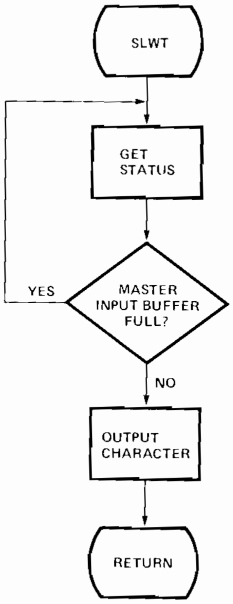


Figure 20. Interface Block Diagram



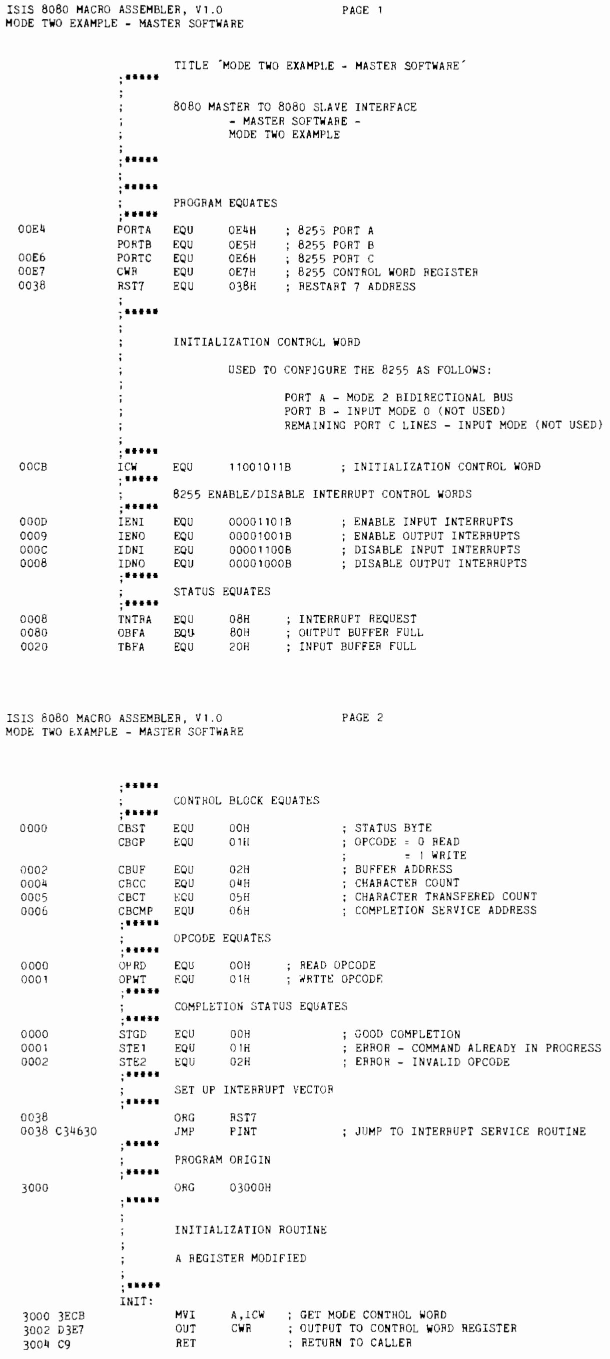
SLAVE READ ROUTINE

```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 1
MODE TWO EXAMPLE - SLAVE SOFTWARE

                        TITLE 'MODE TWO EXAMPLE - SLAVE SOFTWARE'
                ;*****
                ;
                ;       8080 MASTER TO 8080 SLAVE INTERFACE
                ;               - SLAVE SOFTWARE -
                ;               MODE TWO EXAMPLE
                ;
                ;*****
                ;
                ;*****
                ;       PROGRAM EQUATES
                ;*****
 00BF           PDATA   EQU     0BFH    ; INTERPROCESSOR DATA PORT
 007F           PSTS    EQU     07FH    ; STATUS
                ;*****
                ;       BUFFER STATUS MASKS
                ;*****
 0001           OBF     EQU     01H     ; OUTPUT BUFFER FULL
 0002           IBF     EQU     02H     ; INPUT BUFFER FULL
                ;*****
                ;       PROGRAM ORIGIN
                ;*****
 3000                   ORG     03000H
                ;*****
                ;
                ;       SLAVE READ ROUTINE
                ;
                ;               INPUTS:  NONE
                ;               OUTPUTS: CHARACTER READ IN C-REGISTER
                ;
                ;       A,C REG MODIFIED
                ;
                ;*****
                SLRD:
 3000 DB7F              IN      PSTS    ; GET STATUS
 3002 E601              ANI     OBF     ; SEE IF BUFFER FULL
 3004 C20030            JNZ     SLRD    ; NO - LOOP UNTIL FULL
 3007 DBBF              IN      PDATA   ; GET CHARACTER
 3009 4F                MOV     C,A     ; PLACE IN C-REG
 300A C9                RET             ; RETURN TO CALLER
```

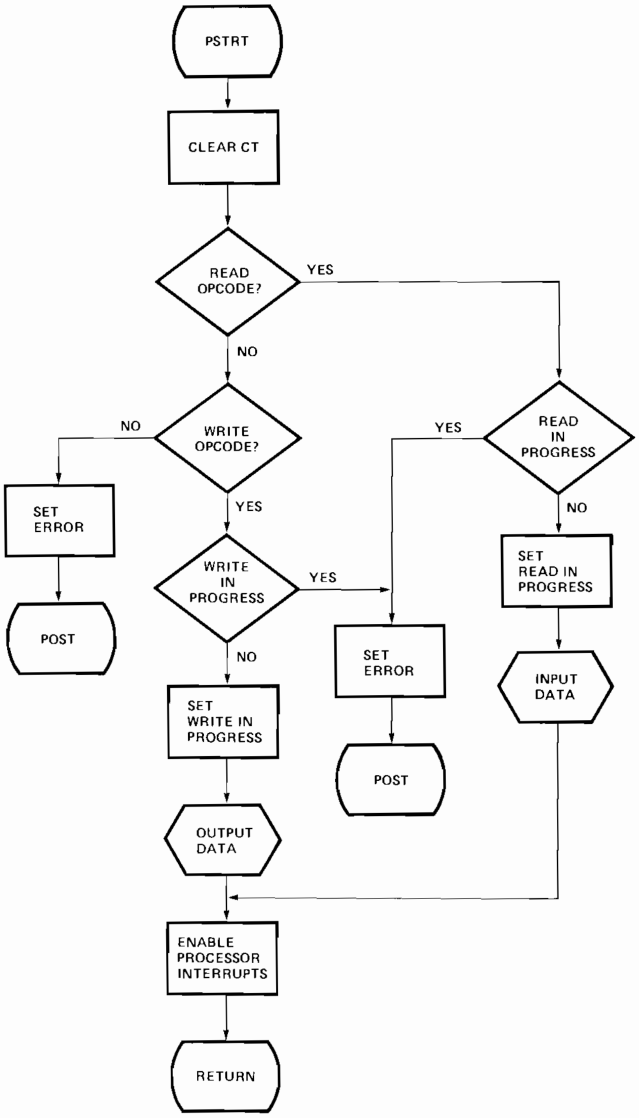


SLAVE WRITE ROUTINE

```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 2
MODE TWO EXAMPLE - SLAVE SOFTWARE

                ;*****
                ;
                ;       SLAVE WRITE ROUTINE
                ;
                ;               INPUTS:  CHARACTER TO WRITE IN C-REGISTER
                ;               OUTPUTS: NONE
                ;
                ;       A REG MODIFIED
                ;
                ;*****
                SLWT:
 300B DB7F              IN      PSTS    ; GET STATUS
 300D E602              ANI     IBF     ; SEE IF BUFFER FULL
 300F C20B30            JNZ     SLWT    ; YES - LOOP UNTIL EMPTY
 3012 79                MOV     A,C     ; GET DATA CHARACTER
 3013 D3BF              OUT     PDATA   ; OUTPUT DATA
 3015 C9                RET             ; RETURN TO CALLER
                ;*****
                ;       END OF SLAVE SOFTWARE DRIVER
                ;*****
 0000                   END
```

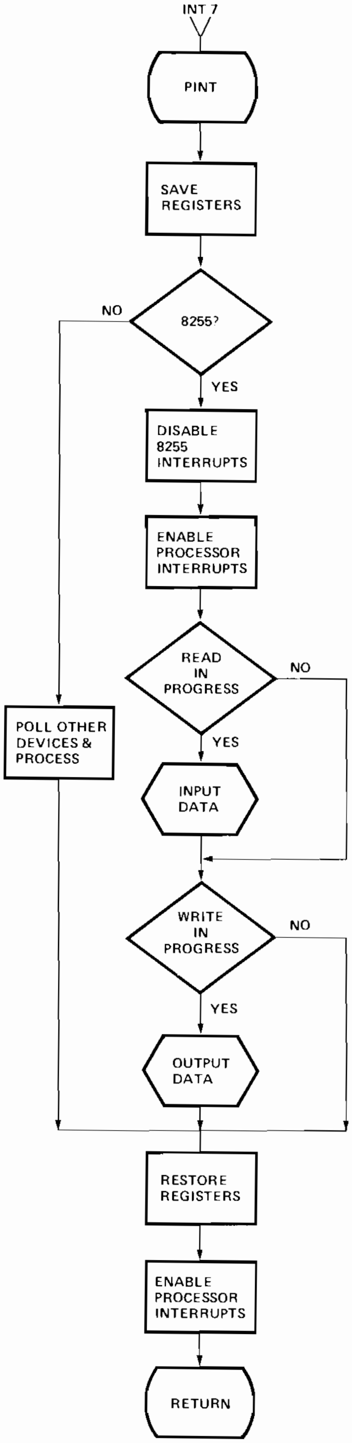


```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 1
MODE TWO EXAMPLE - MASTER SOFTWARE

                        TITLE 'MODE TWO EXAMPLE - MASTER SOFTWARE'
                ;*****
                ;
                ;       8080 MASTER TO 8080 SLAVE INTERFACE
                ;               - MASTER SOFTWARE -
                ;               MODE TWO EXAMPLE
                ;
                ;*****
                ;
                ;*****
                ;       PROGRAM EQUATES
                ;*****
  00E4          PORTA   EQU     0E4H    ; 8255 PORT A
                PORTB   EQU     0E5H    ; 8255 PORT B
  00E6          PORTC   EQU     0E6H    ; 8255 PORT C
  00E7          CWR     EQU     0E7H    ; 8255 CONTROL WORD REGISTER
  0038          RST7    EQU     038H    ; RESTART 7 ADDRESS
                ;
                ;*****
                ;
                ;       INITIALIZATION CONTROL WORD
                ;
                ;               USED TO CONFIGURE THE 8255 AS FOLLOWS:
                ;
                ;                       PORT A - MODE 2 BIDIRECTIONAL BUS
                ;                       PORT B - INPUT MODE 0 (NOT USED)
                ;                       REMAINING PORT C LINES - INPUT MODE (NOT USED)
                ;
                ;*****
  00CB          ICW     EQU     11001011B       ; INITIALIZATION CONTROL WORD
                ;*****
                ;       8255 ENABLE/DISABLE INTERRUPT CONTROL WORDS
                ;*****
  000D          IENI    EQU     00001101B       ; ENABLE INPUT INTERRUPTS
  0009          IENO    EQU     00001001B       ; ENABLE OUTPUT INTERRUPTS
  000C          IDNI    EQU     00001100B       ; DISABLE INPUT INTERRUPTS
  0008          IDNO    EQU     00001000B       ; DISABLE OUTPUT INTERRUPTS
                ;*****
                ;       STATUS EQUATES
                ;*****
  0008          TNTRA   EQU     08H     ; INTERRUPT REQUEST
  0080          OBFA    EQU     80H     ; OUTPUT BUFFER FULL
  0020          TBFA    EQU     20H     ; INPUT BUFFER FULL


ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 2
MODE TWO EXAMPLE - MASTER SOFTWARE


                ;*****
                ;       CONTROL BLOCK EQUATES
                ;*****
  0000          CBST    EQU     00H             ; STATUS BYTE
                CBGP    EQU     01H             ; OPCODE = 0 READ
                                                ;        = 1 WRITE
  0002          CBUF    EQU     02H             ; BUFFER ADDRESS
  0004          CBCC    EQU     04H             ; CHARACTER COUNT
  0005          CBCT    EQU     05H             ; CHARACTER TRANSFERED COUNT
  0006          CBCMP   EQU     06H             ; COMPLETION SERVICE ADDRESS
                ;*****
                ;       OPCODE EQUATES
                ;*****
  0000          OPRD    EQU     00H     ; READ OPCODE
  0001          OPWT    EQU     01H     ; WRTTE OPCODE
                ;*****
                ;       COMPLETION STATUS EQUATES
                ;*****
  0000          STGD    EQU     00H             ; GOOD COMPLETION
  0001          STE1    EQU     01H             ; ERROR - COMMAND ALREADY IN PROGRESS
  0002          STE2    EQU     02H             ; ERROR - INVALID OPCODE
                ;*****
                ;       SET UP INTERRUPT VECTOR
                ;*****
  0038                  ORG     RST7
  0038 C34630           JMP     PINT            ; JUMP TO INTERRUPT SERVICE ROUTINE
                ;*****
                ;       PROGRAM ORIGIN
                ;*****
  3000                  ORG     03000H
                ;*****
                ;
                ;       INITIALIZATION ROUTINE
                ;
                ;       A REGISTER MODIFIED
                ;
                ;*****
                INIT:
  3000 3ECB             MVI     A,ICW   ; GET MODE CONTHOL WORD
  3002 D3E7             OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
  3004 C9               RET             ; RETURN TO CALLER
```



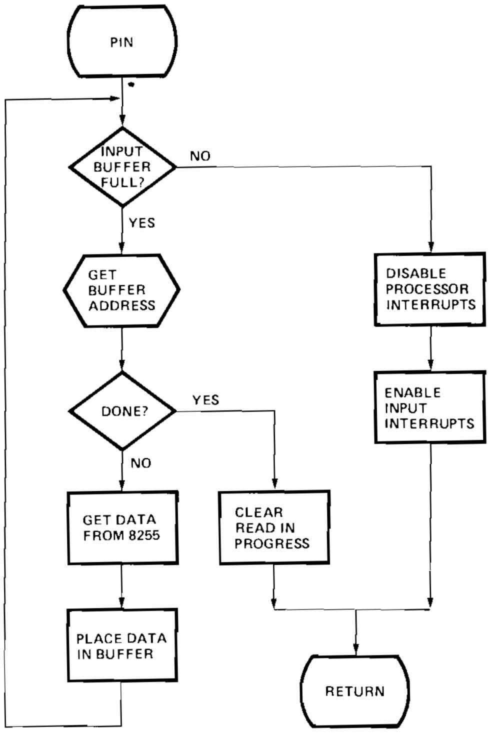
```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 3
COMMAND PROCESSOR


                 ;*****
                 ;
                 ;       COMMAND PROCESSOR
                 ;
                 ;               INPUTS: CONTROL BLOCK ADDRESS IN D AND E REGIST:RS
                 ;
                 ;               OUTPUTS: START I/O OR ERROR STATUS IN CONTROL B.OCK
                 ;       A,H,L   MODIFIED
                 ;
                 ;*****
                 PSTRT:
 3005 210500             LXI     H,CBCT  ; GET INDEX TO CT
 3008 19                 DAD     D       ; COMPUTE ADDRESS OF CT
 3009 3600               MVI     M,OPRD  ; CLEAR CT
 300B 210100             LXI     H,CBOP  ; GET INDEX TO OPCODE
 300E 19                 DAD     D       ; COMPUTE ADDRESS
                         MOV     A,M     ; GET OPCODE
 3010 FE00               CPI     00H     ; SEE IF READ
 3012 CA2430             JZ      PSRD    ; YES - GO PROCESS READ
 3015 FE01               CPI     OPWT    ; SEE IF WRITE
 3017 CA3530             JZ      PSWT    ; YES - GO PROCESS WRITE
                 ;*****
                 ;       ERROR - INVALID OPCODE
                 ;*****
 301A 3E02               MVI     A,STE2  ; GET ERROR STATUS CODE
                         JMP     POST    ; CALL COMPLETION ROUTINE
                 ;*****
                 ;       ERROR - TRANSACTION ALREADY IN PROGRESS
                 ;*****
                 PSTE:
 301F 3E01               MVI     A,STE1  ; GET ERROR STATUS CODE
 3021 C3DC30             JMP     POST    ; CALL COMPLETION ROUTINE
                 ;*****
                 ;       PROCESS READ COMMAND
                 ;*****
                 PSRD:
 3024 3AEA3D             LDA     PRGRD+1 ; GET READ IN PROGRESS ADDRESS
 3027 A7                 ANA     A       ; SEE IF READ IN PROGRESS (TEST FOR ZER( )
 3028 C21F30             JNZ     PSTE    ; IF YES - BRANCH
 302B EB                 XCHG
 302C 22E930             SHLD    PRGRD   ; SAVE CONTROL BLOCK ADDRESS
 302F EB                 XCHG
 3030 CD7C30             CALL    PIN     ; START I/O
 3033 FB                 EI              ; ENABLE INTERRUPTS
 3034 C9                 RET             ; RETURN TO CALLER
```

```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 4
COMMAND PROCESSOR


                 ;*****
                 ;       PROCESS WRITE COMMAND
                 ;*****
                 PSWT:
 3035 3AEC30             LDA     PRGWT+1 ; GET WRITE IN PROGRESS ADDRESS
 3038 A7                 ANA     A       ; SEE IF WRITE IN PROGRESS (TEST FOR ZER))
 3039 C21F30             JNZ     PSTE    ; IF YES - BRANCH
 303C EB                 XCHG
 303D 22EB30             SHLD    PRGWT   ; SAVE CONTROL BLOCK ADDRESS
 3040 EB                 XCHG
 3041 CD9C30             CALL    POUT    ; START I/O
 3044 FB                 EI              ; ENABLE INTERRUPTS
 3045 C9                 RET             ; RETURN TO CALLER
```

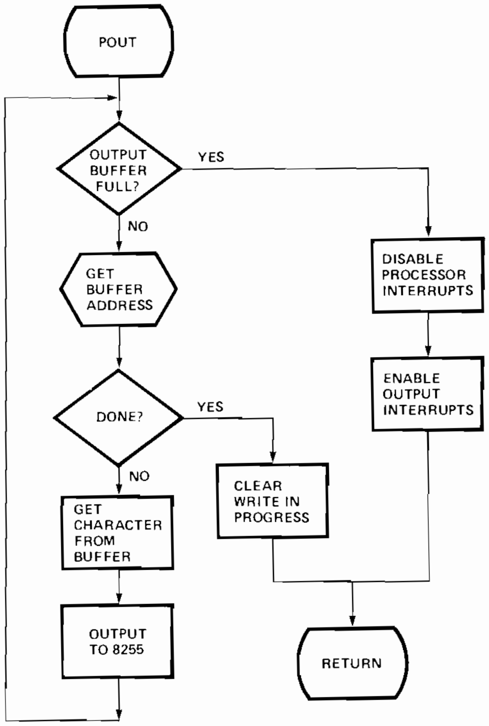


```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 5
INTERRUPT SERVICE ROUTINE


                 ;*****
                 ;
                 ;         INTERRUPT SERVICE ROUTINE
                 ;        ALL REGISTERS SAVED AND RESTORED
                 ;
                 ;*****
                 PINT:
  3046 F5                PUSH    PSW      ; SAVE PSW
  3047 C5                PUSH    B        ; SAVE REGISTER PAIR B AND C
  3048 D5                PUSH    D        ; SAVE REGISTER PAIR D AND E
                         PUSH    H        ; SAVE REGISTER PAIR H AND L
                 ;*****
                 ;       POLL INTERRUPT SOURCE - SEE IF 8255
                 ;*****
  304A DBE6              IN      PORTC    ; GET STATUS OF DEVICE
  304C E608              ANI     INTRA    ; SEE IF INT
  304E CA7630            JZ      PPOLL    ; NO - BRANCH TO POLL CTHER DEVICES IF ANY
  3051 3E0C              MVI     A,IDNI   ; GET INPUT INT DISABLE CONTROL WORD
  3053 D3E7              OUT     CWR      ; DISABLE DEVICE INTERRUPTS
  3055 3E08              MVI     A,IDNO   ; GET OUTPUT INT DISABLE CONTROL WORD
  3057 D3E7              OUT     CWR      ; DISAPE DEVICE INTERRUPTS
  3059 FB                EI               ; ENABLE PROCESSOR INTERRUPTS
  305A 2AE930            LHLD    PRGRD    ; GET READ CONTROL BLOCK
  305D AF                XRA     A        ; CLEAR A REG
  305E BC                CMP     H        ; SEE IF READ IN PROGRESS
  305F CA6530            JZ      PINT1    ; NO - BRANCH
  3062 CD7C30            CALL    PIN      ; DO INPUT
                 PINT1:
  3065 2AEB30            LHLD    PRGWT    ; GET WRITE CONTROL BLOCK
  3068 AF                XRA     A        ; CLEAR A REG
  3069 BC                CMP     H        ; SEE IF WRITE IN PROGRESS
  306A CA7030            JZ      PRTN     ; NO - BRANCH
  306D CD9C30            CALL    POUT     ; DO OUTPUT
                 ;*****
                 ;       RESTORE REGISTERS AND RETURN FROM INTERRUPT
                 ;*****
                 PRTN:
  3070 E1                POP     H        ; RESTORE REGISTER PAIR H AND L
  3071 D1                POP     D        ; RESTORE REGISTER PAIR D AND E
  3072 C1                POP     B        ; RESTORE REGISTER PAIR B AND C
  3073 F1                POP     PSW      ; RESTORE PSW
  3074 FB                EI               ; ENABLE PROCESSOR INTERRUPTS
  3075 C9                RET              ; RETURN TO INTERRUPTED PROCESS
```

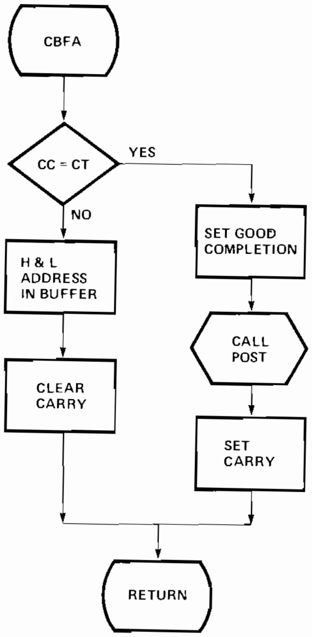
```
ISIS 8080 MACRO ASSEMBLER, V1.0                PAGE 6
INTERRUPT SERVICE ROUTINE


                 ;*****
                 ;
                 ;       POLL OTHER DEVICES IF ANY
                 ;               IF NO OTHER DEVICES TO POLL - USER SUPPLIED ERROR
                 ;               RECOVERY ROUTINE.
                 ;
                 ;*****
                 PPOLL:
  3076 C37030            JMP     PRTN     ; RETURN
                 ;*****
                 ;       ERROR - INTERRUPT FROM IDLE DEVICE
                 ;               USER SUPPLIED ERROR RECOVERY ROUTINE
                 ;*****
                 PIER1:
  3079 C37030            JMP     PRTN     ; RETURN
```



```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 7
INPUT DATA ROUTINE


                ;*****
                ;       INPUT DATA ROUTINE
                ;*****
                PIN:
 307C DBE6              IN      PORTC   ; GET STATUS OF DEVICE
 307E E620              ANI     IBFA    ; SEE IF INPUT BUFFER FULL
 3080 CA9630            JZ      PRTI    ; NO - BRANCH
 3083 CDBC30            CALL    CBFA    ; GET ADDRESS IN BUFFER
 3086 DA8F30            JC      PIDON   ; IF DONE - BRANCH
 3089 DBE4              IN      PORTA   ; GET DATA
 308B 77                MOV     M,A     ; PLACE IN BUFFER
 308C C37C30            JMP     PIN     ; LOOP
                ;*****
                ;       END OF INPUT TRANSACTION
                ;*****
                PIDON:
 308F AF                XRA     A       ; CLEAR A
 3090 32EA30            STA     PRGRD+1 ; CLEAR READ IN PROGRESS
 3093 C39630            JMP     PRTI    ; RETURN
                ;*****
                ;       RETURN FROM INPUT
                ;*****
                PRTI:
 3096 F3                DI              ; DISABLE PROCESSOR INTERRUPTS
 3097 3E0D              MVI     A,IENI  ; GET ENABLE INPUT INTERRUPTS CONTROL WOI D
 3099 D3E7              OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
 309B C9                RET             ; RETURN TO CALLER
```



```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 8
OUTPUT DATA ROUTINE


                ;*****
                ;       OUTPUT DATA ROUTINE
                ;*****
                POUT:
 309C DBE6              IN      PORTC   ; GET PORTC STATUS
 309E E620              ANI     IBFA    ; SEE IF OUTPUT BUFFER FULL
                        JNZ     PRTO    ; YES - BRANCH
 30A3 CDBC30            CALL    CBFA    ; SET UP ADDRESS OF DATA
 30A6 DAAF30            JC      PODON   ; IF DONE - BRANCH
 30A9 7E                MOV     A,M     ; GET DATA FROM BUFFER
 30AA D3E4              OUT     PORTA   ; OUTPUT DATA
 30AC C39C30            JMP     POUT    ; LOOP
                ;*****
                ;       END OF OUTPUT TRANSACTION
                ;*****
                PODON:
 30AF AF                XRA     A       ; CLEAR A REG
 30B0 32EC30            SIA     PRGWT+! ; CLEAR WRITE IN PROGRESS
 30B3 C3B630            JMP     PRTO    ; RETURN
                ;*****
                ;       RETURN FROM OUTPUT
                ;*****
                PRTO:
 30B6 F3                DI              ; DISABLE PROCESSOR INTERRUPTS
 30B7 3E09              MVI     A,IENO  ; GET ENABLE OUTPUT INTERRUPTS CONTROL WORD
 30B9 D3E7              OUT     CWR     ; OUTPUT TO CONTROL WORD REGISTER
 30BB C9                RET             ; RETURN TO CALLER
```

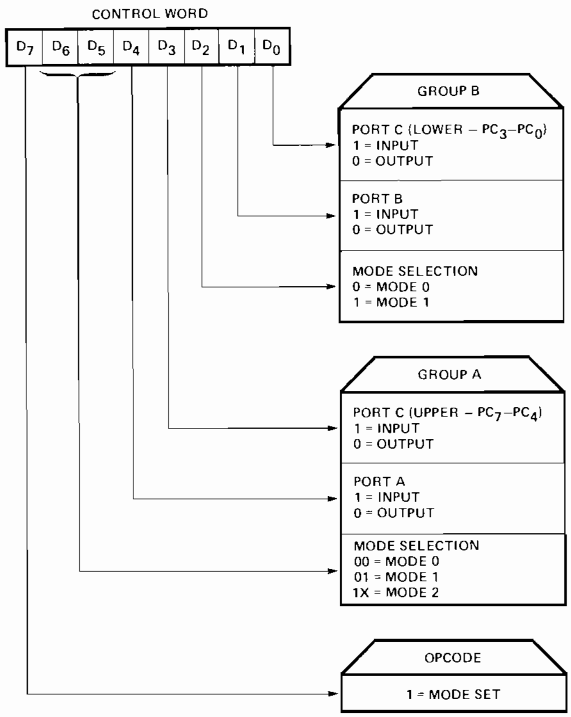


Setup Buffer Address Subroutine

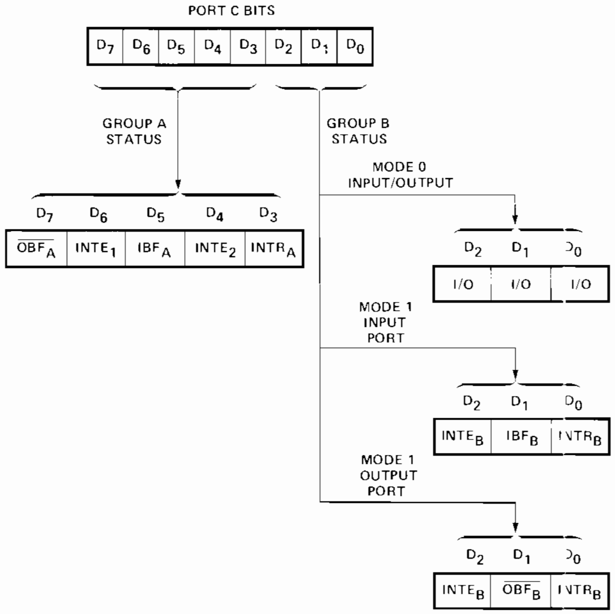
```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 9
COMPUTE BUFFER ADDRESS ROUTINE


                ;*****
                ;       COMPUTE BUFFER ADDRESS ROUTINE
                ;*****
                CBFA:
 30BC 210500            LXI     H,CBCT  ; GET INDEX TO CT
 30BF 19                DAD     D       ; COMPUTE ADDRESS OF CT
 30C0 7E                MOV     A,M     ; GET CT
 30C1 34                INR     M       ; INC CT
 30C2 2B                DCX     H       ; DEC TO CC
 30C3 BE                CMP     M       ; SEE IF EQUAL
 30C4 CAD530            JZ      PCOMP   ; IF EQUAL - DONE GO TELL USER
 30C7 210200            LXI     H,CBUF  ; GET INDEX TO BUFFER ADDRESS
 30CA 19                DAD     D       ; COMPUTE ADDRESS OF BUFFER ADDRESS
 30CB D5                PUSH    D       ; SAVE D AND E REGISTERS
 30CC 5E                MOV     E,M     ; GET LSB OF BUFFER ADDRESS
 30CD 23                INX     H       ; INC TO NEXT BYTE
 30CE 56                MOV     D,M     ; GET BUFFER MSB
 30CF AC                XRA     H       ; CLEAR H REG
 30D0 6F                MOV     L,A     ; GET CT
 30D1 19                DAD     D       ; COMPUTE CHARACTER ADDRESS
 30D2 D1                POP     D       ; RESTORE CONTROL BLOCK ADDRESS
 30D3 AF                XRA     A       ; CLEAR CARRY
 30D4 C9                RET             ; RETURN TO CALLER
```

```
ISIS 8080 MACRO ASSEMBLER, V1.0                 PAGE 10
POST TO USER COMPLETION ROUTINE


                ;*****
                ;       POST GOOD COMPLETION TO USER
                ;*****
                PCOMP:
 30D5 3E00              MVI     A,SIGD  ; GET GOOD STATUS CODE
 30D7 CDDC30            CALL    POST    ; CALL USER ROUTINE
 30DA 37                STC             ; SET CARRY
 30DB C9                RET             ; RETURN TO CALLER
                ;*****
                ;       POST TO USER COMPLETION ROUTINE
                ;
                ;               INPUTS : STATUS CODE IN A REG
                ;                        CONTROL BLOCK ADDRESS IN D AND E REG
                ;               OUTPUTS: PASSES CONTROL TO USER COMPLETION ADDRESS
                ;                        SPECIFIED IN CONTROL BLOCK
                ;*****
                POST:
 30DC EB                XCHG
 30DD 77                MOV     M,A     ; UPDATE STATUS
 30DE EB                XCHG
 30DF 210600            LXI     H,CBCMP ; GET INDEX TO COMPLETION ADDRESS
 30E2 19                DAD     D       ; COMPUTE ADDRESS
 30E3 4E                MOV     C,M     ; GET LSB OF COMPLETION ADDRESS
 30E4 23                INX     H       ; INC TO NEXT BYTE
 30E5 46                MOV     B,M     ; GET MSB BYTE OF COMPLETION ADDRESS
 30E6 C5                PUSH    B       ; PUSH ADDRESS INTO STACK
 30E7 C9                RET             ; PASS CONIROL TO USER ROUTINE
 30E8 C9                RET             ; REIURN TO CALLER
                ;*****
                ;
                ;       DATA AND TABLES
                ;
                ;               IF DATA NON ZERO CONTROL BLOCK IN PROGRESS
                ;
                ;*****

 30E9 0000      PRGRD:  DW      0       ; IN PROGRESS READ CONTROL BLOCK
 30EB 0000      PRGWT:  DW      0       ; IN PROGRESS WRITE CONTROL BLOCK
                ;*****
                ;       END OF MASTER SOFTWARE DRIVER
                ;*****
 0000                   END
```

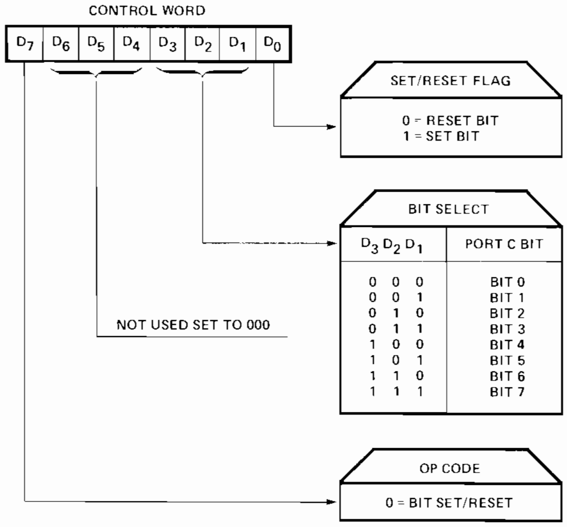
APPENDIX A – 8255 QUICK REFERENCE



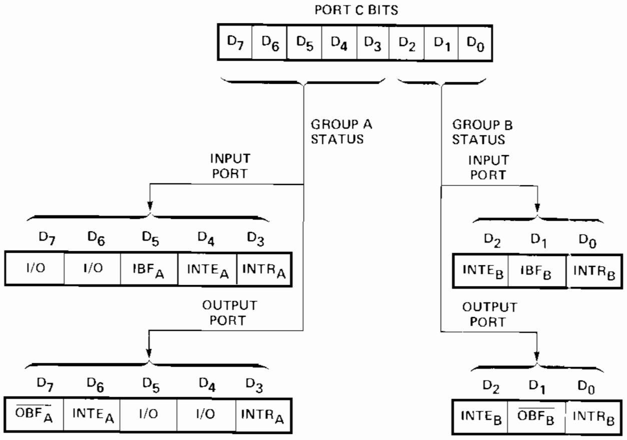
MODE CONTROL WORD



MODE 1 STATUS WORD

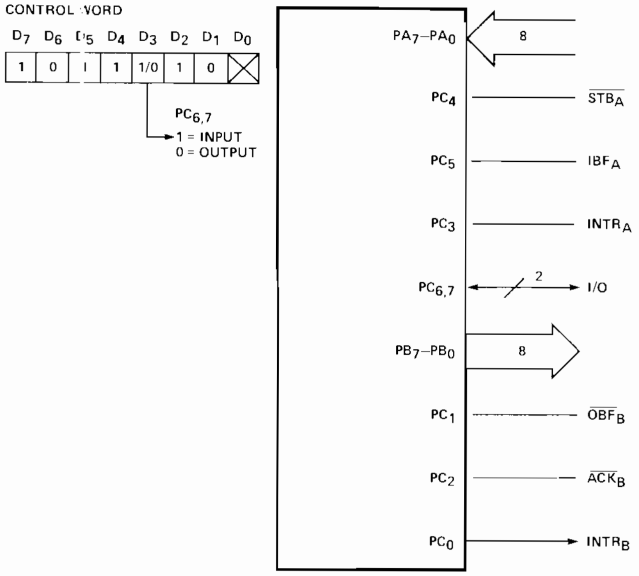


BIT SET/RESET CONTROL WORD

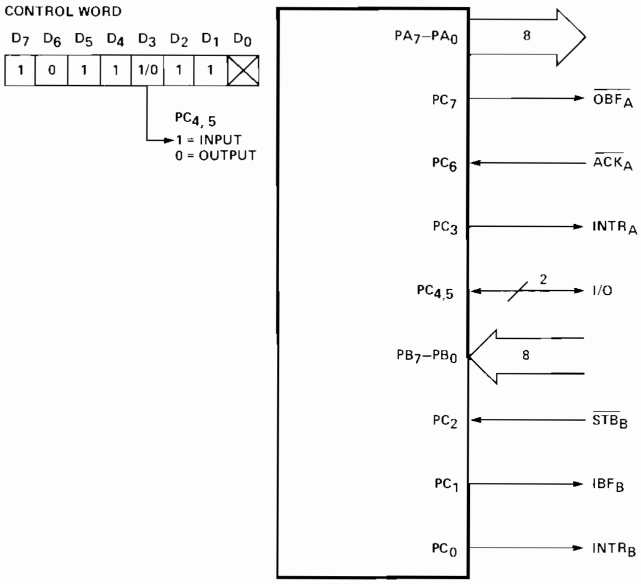


MODE 2 STATUS WORD

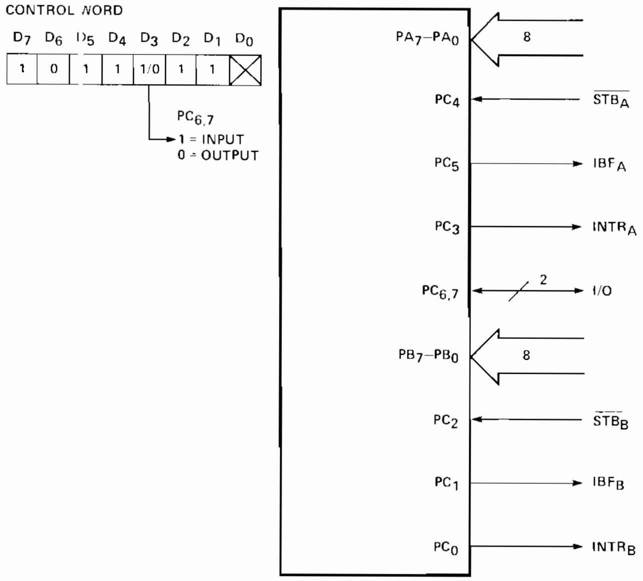
MODE 1 CONFIGURATIONS



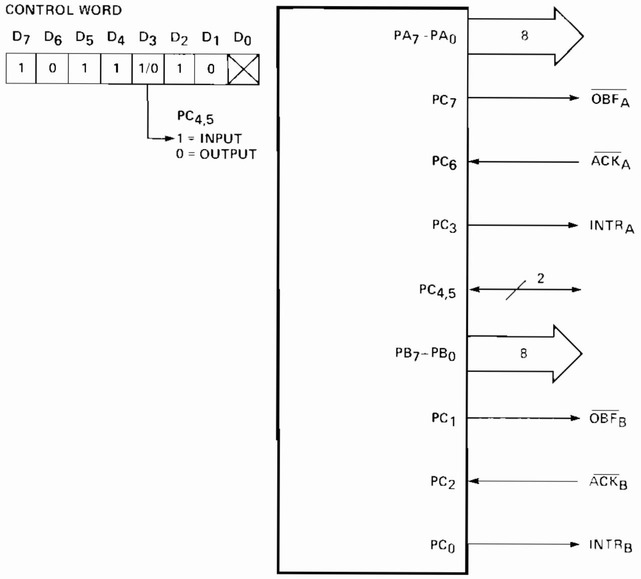
PORT A – STROBED INPUT
PORT B – STROBED OUTPUT



PORT A – STROBED OUTPUT
PORT B – STROBED INPUT

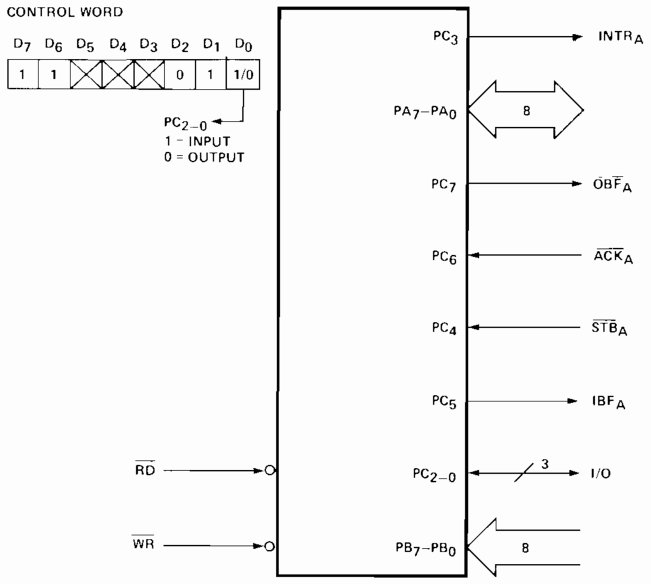


PORT A – STROBED INPUT
PORT B – STROBED INPUT

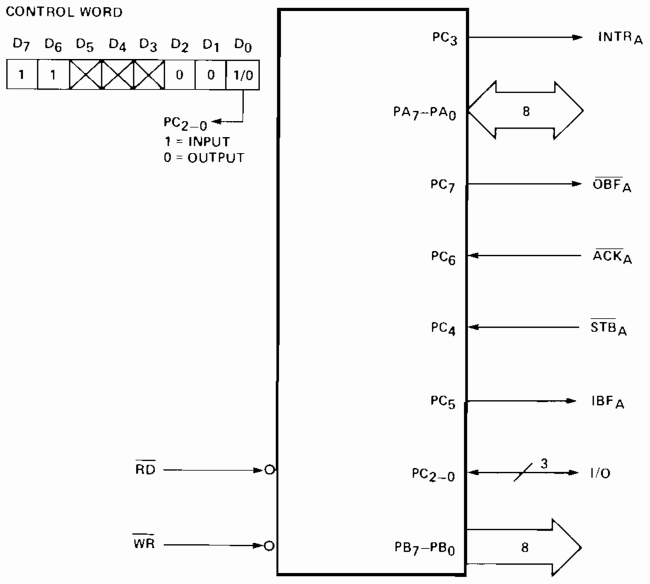


PORT A – STROBED OUTPUT
PORT B – STROBED OUTPUT

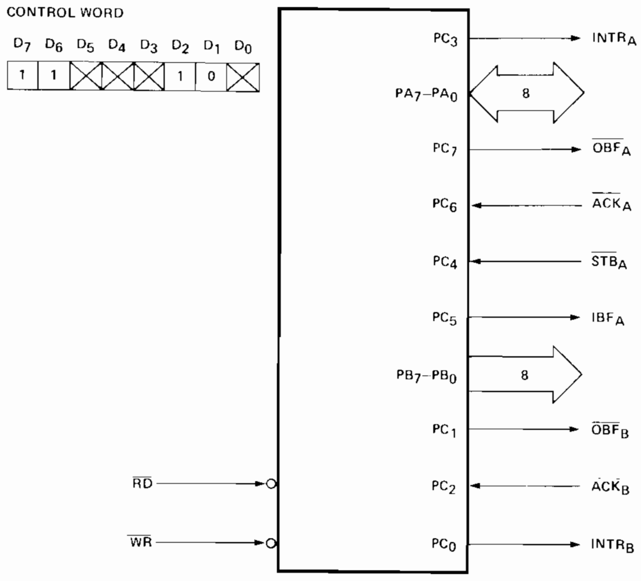
MODE 2 CONFIGURATIONS



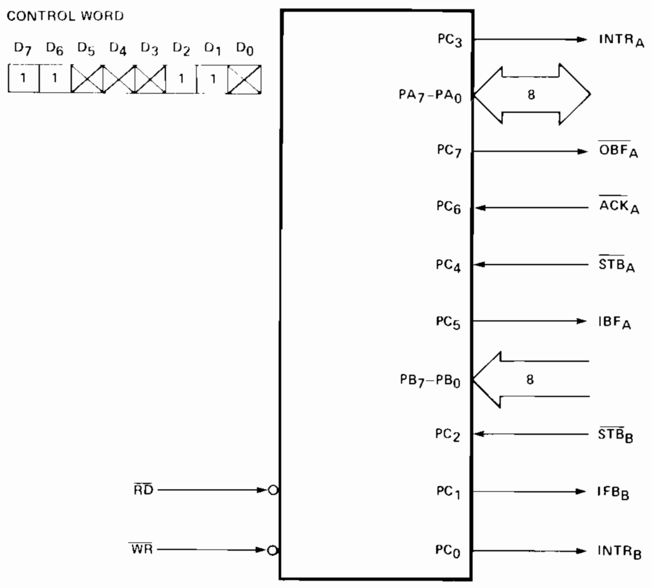
PORT A – MODE 2
PORT B – MODE 0 INPUT



PORT A – MODE 2
PORT B – MODE 0 INPUT



PORT A – MODE 2
PORT B – MODE 1 OUTPUT



PORT A – MODE 2
PORT B – MODE 1 OUTPUT

U.S. AND CANADIAN SALES OFFICES

ALABAMA
Barnhill and Associates
7844 Horseshoe Trail
Huntsville 35802
Tel: (205) 883-9394

ARIZONA
Sales Engineering, Inc.
7155 E. Thomas Road, No. 6
Scottsdale 85252
Tel: (602) 945-5781
TWX: 910-950-1288

CALIFORNIA
Intel Corp.*
990 E. Arques Ave.
Suite 112
Sunnyvale 94086
Tel: (408) 738-3870
TWX: 910-339-9279

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Earle Associates, Inc.
4433 Convoy Street
Suite A
San Diego 92111
Tel: (714) 278-5441
TWX: 910-335-1585

Intel Corp.*
1651 East 4th Street
Suite 228
Santa Ana 92701
Tel: (714) 835-9642
TWX: 910-595-1114

COLORADO
Intel Corp.
12075 East 45th Avenue
Suite 310
Denver 80239
Tel: (303) 373-4920
TWX: 910-932-0322

FLORIDA
Intel Corp.
1090 NE 27th Terrace
Pompano Beach 33062
Tel: (305) 781-7450
TWX: 510-956-9407

FLORIDA (cont.)
Intel Corp.
5151 Adanson Street, Suite 200-3
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

ILLINOIS
Intel Corp.*
900 Jorie Boulevard
Suite 138
Oakbrook 60521
Tel: (312) 325-9510
TWX: 910-651-5881

IOWA
Technical Representatives, Inc.
1703 Hillside Drive
Cedar Rapids
Tel: (319) 396-5662

KANSAS
Technical Representatives, Inc.
801 Clairborne
Olathe 66061
Tel (913) 782-1177
TWX: 910-749-6412

MARYLAND
Barnhill and Associates
57 West Timonium Road
Timonium 21093
Tel: (301) 252-7742

Intel Corp.*
57 West Timonium Road
Suite 307
Timonium 21093
Tel: (301) 252-7742
TWX: 710-232-1807

MASSACHUSETS
Datcom
55 Moody Street
Waltham 02154
Tel: (617) 891-4600
TELEX: 92-3462

Intel Corp.*
187 Billerica Road, Suite 14A
Chelmsford 01824
Tel: (617) 861-1136
TWX: 710-343-6333

MICHIGAN
Intel Corp.
725 South Adams Road
Suite 288
Birmingham 48011
Tel: (313) 642-7018
TWX: 910-420-1212
TELEX: 2 31143

MINNESOTA
Intel Corp.
675 Southgate Office Plaza
5001 West 80th Street
Bloomington 55437
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI
Technical Representatives, Inc.
Trade Center Bldg.
300 Brookes Drive, Suite 108
Hazelwood 63042
Tel: (314) 731-5200
TWX: 910-762-0618

NEW JERSEY
Intel Corp.
2 Kilmer Road
Edison 08817
Tel: (201) 985-9100
TWX: 710-480-6238

NEW YORK
Intel Corp.*
6901 Jericho Turnpike
Syosset 11791
Tel: (516) 364-9860
TWX: 510-221-2198

Intel Corp.
474 Thurston Road
Rochester, N.Y. 14619
Tel: (716) 328-7340
TWX: 510-253-3841

T-Squared
3522 James Street
Syracuse 13206
Tel: (315) 463-8592
TWX: 710 541-0554

NEW YORK (cont.)
T-Squared
640 Craig Road
P.O. Box W
Pittsford 14534
Tel: (716) 381-2551
TELEX: 97-8289

Intel Corp.
55 Market Street
Poughkeepsie, New York 12601
Tel: (914) 473-2303
TWX: 510-248-0060

NORTH CAROLINA
Barnhill and Associates
913 Plateau Lane
Raleigh 27609
Tel: (919) 876-5617

OHIO
Intel Corp.*
8312 North Main Street
Dayton 45415
Tel: (513) 890-5350
TELEX: 288-004

Intel Corp.*
27801 Euclid Ave.
Suite 450
Euclid 44132
Tel: (216) 289-0101

PENNSYLVANIA
Vantage Sales Company
21 Bala Avenue
Bala Cynwyd 19004
Tel: (215) 667-0990
TWX: 510-662-5846

Intel Corp.*
1777 Walton Rd.
Suite 328A
Blue Bell 19422
Tel: (215) 542-9444
TWX: 510-661-0709

TENNESSEE
Barnhill and Associates
206 Chickasaw Drive
Johnson City 37601
Tel: (615) 928-0184

TEXAS
Evans & McDowell Associates
13777 N. Central Expressway
Suite 405
Dallas 75231
Tel: (214) 238-7157
TWX: 910-867-4763

Evans & McDowell Associates
6610 Harwin Avenue, Suite 125
Houston 77036
Tel: (713) 783-2900

Intel Corp.*
6350 L.B.J. Freeway
Suite 178
Dallas 75240
Tel: (214) 661-8829
TWX: 910-860-5487

VIRGINIA
Barnhill and Associates
P.O. Box 1104
Lynchburg 24505
Tel: (804) 846-4624

WASHINGTON
E.S./Chase Co.
P.O. Box 80903
Seattle 98108
Tel: (206) 762-4824
Twx: 910-444-2298

CANADA
Multilek, Inc.
4 Barren Street
Ottawa, Ontario K2J 1G2
Tel: (613) 825-4695
TELEX: 053-4585

EUROPEAN MARKETING OFFICES

BELGIUM
Intel International*
Rue du Moulin à Papier
51-Boite 1
B-1160 Brussels
Tel: (02) 660 30 10
TELEX: 24814

FRANCE
Intel Corporation, S.A.R.L.*
74, Rue D'Arcueil
Silic 223
94528 Rungis Cedex
Tel: (01) 687 22 21
TELEX: 270475

SCANDINAVIA
Intel Scandinavia A/S*
Lyngbyvej 32 2nd Floor
DK-2100 Copenhagen East
Denmark
Tel: (01) 18 20 00
TELEX: 19567

Intel Sweden AB*
Box 86
S-16212 Vällingby 1
Sweden
Tel: (08) 37 53 70
TELEX: 13164 (ABCENT)

ENGLAND
Intel Corporation (U.K.) Ltd.*
Broadfield House
4 Between Towns Road
Cowley, Oxford OX4 3NB
Tel: (0865) 77 14 31
TELEX: 837203

Intel Corporation (U.K.) Ltd.
46-50 Beam Street
Nantwich, Cheschire CW5 5LJ
Tel: (0270) 62 65 60
TELEX: 36620

GERMANY
Intel Semiconductor GmbH*
Wolfratshauserstrasse 169
D8 Munich 71
Tel: (089) 79 89 23
TELEX: 5-212870

Intel Semiconductor GmbH
D-6272 Niedernhausen
Wiesenweg 26
Tel: (06127) 2314
TELEX: 04186183

Intel Semiconductor GmbH
D-7000 Stuttgart 80
Ernsthaldenstrasse 17
Tel: (0711) 7351506
TELEX: 7255346

ORIENT MARKETING OFFICES

JAPAN
Intel Japan Corporation*
Flower Hill-Shinmachi East Bldg.
1-23-9, Shinmachi, Setagaya-ku
Tokyo 154
Tel: (03) 426-9261
TELEX: 781-28426

HONG KONG
Q1 (Far East) Ltd.
Tak Yan Commercial Bldg. 6th floor
30-32 D'Aguilar Street, Central
Hong Kong
Tel: 5-260311
TELEX: 83138 JADE HX

TAIWAN
Taiwan Automation Co.*
8th Floor, 140, Section 1
Chung Hsiao E. Road
Taipei
Tel: 393-1115
TELEX: 11942 TAIAUTO

TAIWAN (cont.)
Asionics-Taiwan, Inc.
205 Pa-Teh Road, Section 4
Taipei
Tel: 75 55 82
TELEX: 22158 Asionics

INTERNATIONAL DISTRIBUTORS

AUSTRALIA
A. J. Ferguson (Adelaide) PTY, Ltd.
44 Prospect Rd.
Prospect 5082
South Australia
Tel: 269-1244
TELEX: 82635

AUSTRIA
Bacher Elektronische Gerate GmbH
Meidlinger Hauptstrasse 78
A 1120 Vienna
Tel: (0222) 83 63 96
TELEX: (01) 1532

BELGIUM
Inelco Belgium S.A.
Avenue Val Duchesse, 3
B-1160 Brussels
Tel: (02) 660 00 12
TELEX: 25441

DENMARK
Scandinavian Semiconductor
Supply A/S
Nannasgade 18
DK-2200 Copenhagen N
Tel: (01) 93 50 90
TELEX: 19037

FINLAND
Oy Fintronic AB
Karjalankatu 2C,
SF 00520
Helsinki 52
Tel: (90) 664 451
TELEX: 12426

FRANCE
Tekelec Airtronic
Cite des Bruyeres
Rue Carle Vernet
92310 Sevres
Tel: (1) 027 75 35
TELEX: 250997

GERMANY
Alfred Neye Enatachnik GmbH
Schillerstrasse 14
D-2085 Quickborn-Hamburg
Tel: (04106) 6121
TELEX: 02-13590

Electronic 2000 Vertriebs GmbH
Neumarketer Strasse 75
D-8000 Muenchen 80
Tel: (089) 434061
TELEX: 484426

Jermyn GmbH
Postfach 1146
D-6277 Kamberg
Tel: (06434) 6005
TELEX: 484426

HONG KONG
ASTEC International
Keystone House, 2nd Floor
Hankow Road, Kowloon
Tel: 3-687760
TELEX: 74899 ASCOM

ISRAEL
Telsys Ltd.
54, Jabotinsky Road
IL-Ramat - Gan 52 464
Tel: (3) 73 98 65
TELEX: 32392

Eastronics Ltd.
11 Rozanis Street
P.O. Box 39300
Tel-Aviv
Tel: 475151
TELEX: 33638

ITALY
Eledra 3S S.P.A.
Viale Elvezia, 18
20154 Milan,
Tel: (02) 3493041
TELEX: 39332

Eledra 3S S.P.A.
Via Giuseppe Valmarana, 63
00139 Rome, Italy
Tel: (06) 81 27 290 - 81 27 324

JAPAN
Pan Electron
No. 1 Higashikata-Machi
Midori-Ku, Yokohama 226
Tel: (045) 471-8811
TELEX: 781-4773

NETHERLANDS
Inelco Nederland
AFD Elektronic
Joan Muyskenweg 22
NL-1006 Amsterdam
Tel: (020) 934824
TELEX: 14622

NORWAY
Nordisk Elektronik (Norge) A/S
Mustads Vei 1
N-Oslo 2
Tel: (02) 55 38 93
TELEX: 16963

SOUTH AFRICA
Electronic Building Elements
P.O. Box 4609
Pretoria
Tel: 78 92 21
TELEX: 30181

SPAIN
Interface
Ronda General Mitre #7
E-Barcelona 17
Tel: (93) 203-53-30
TELEX: 52838

SWEDEN
Nordisk Electronik AB
Fack
S-10380 Stockholm 7
Tel: (08) 248340
TELEX: 10547

SWITZERLAND
Industrade AG
Gemsenstrasse 2
Postcheck 80 - 21190
CH-8021 Zurich
Tel: (01) 60 22 30
TELEX: 56788

UNITED KINGDOM
Rapid Recall, Ltd.
11-15 Betterton Street
Drury Lane
London WC2H 9BS
Tel: (01) 379-6741
TELEX: 28752

G.E.C. Semiconductors Ltd
East Lane
Wembley HA9 7PP
Middlesex
Tel: (01) 904-9303
TELEX: 923429

Jermyn Industries
Bestry, Sevenoaks Road
Sevenoaks, Kent.
Tel: (0732) 51174
TELEX: 95143

***Field Application Location**

U.S. AND CANADIAN DISTRIBUTORS

ALABAMA

†Hamilton/Avnet Electronics
805 Oser Drive NW
Huntsville 35805
Tel: (205) 533-1170

ARIZONA

Cramer/Arizona
2643 East University Drive
Phoenix 85034
Tel: (602) 263-1112

Hamilton/Avnet Electronics
2615 South 21st Street
Phoenix 85034
Tel: (602) 275-7851

Liberty/Arizona
3130 N. 27th Avenue
Phoenix 85107
Tel: (602) 257-1272
TELEX: 910-951-4282

CALIFORNIA

†Hamilton/Avnet Electronics
575 E. Middlefield Road
Mountain View 94040
Tel: (415) 961-7000

†Hamilton/Avnet Electronics
8917 Complex Drive
San Diego 92123
Tel: (714) 279-2421

†Hamilton Electro Sales
10912 W. Washington Boulevard
Culver City 90230
Tel: (213) 558-2121

†Cramer/San Francisco
720 Palomar Avenue
Sunnyvale 94086
Tel: (408) 739-3011

†Cramer/Los Angeles
1720 Daimler Street
Irvine 92705
Tel: (714) 979-3000

Cramer/San Diego
8975 Complex Drive
San Diego 92123
Tel: (714) 565-1881

†Liberty Electronics
124 Maryland Street
El Segundo 90245
Tel: (213) 322-8100
Tel: (714) 638-7601
TWX: 910-348-7140

Liberty/San Diego
8248 Mercury Court
San Diego 92111
Tel: (714) 565-9171
TELEX: 910-335-1590

Elmar Electronics
2288 Charleston Road
Mountain View 94040
Tel: (415) 961-3611
TELEX: 910-379-6437

COLORADO

Cramer/Denver
5465 E. Evans Pl. at Hudson
Denver 80222
Tel: (303) 758-2100

Elmar/Denver
6777 E. 50th Avenue
Commerce City 80022
Tel: (303) 287-9611
TWX: 910-936-0770

†Hamilton/Avnet Electronics
5921 No. Broadway
Denver 80216
Tel: (303) 534-1212

CONNECTICUT

Cramer/Connecticut
35 Dodge Avenue
North Haven 06473
Tel: (203) 239-5641

Components Plus
361 W. State
Westport 08880
Tel: (203) 226-4731

Hamilton/Avnet Electronics
643 Danbury Road
Georgetown 06829
Tel: (203) 762-0361

FLORIDA

Cramer/E.W. Hollywood
4035 No. 29th Avenue
Hollywood 33020
Tel: (305) 923-8181

†Hamilton/Avnet Electronics
4020 No. 29th Ave.
Hollywood 33021
Tel: (305) 925-5401

†Cramer/EW Orlando
345 No. Graham Ave.
Orlando 32814
Tel: (305) 894-1511

GEORGIA

Cramer/EW Atlanta
3923 Oakcliff Industrial Center
Atlanta 30340
Tel: (404) 448-9050

Hamilton/Avnet Electronics
6700 I 85, Access Road, Suite 2B
Norcross 30071
Tel: (404) 448-0800

ILLINOIS

†Cramer/Chicago
1911 So. Busse Rd.
Mt. Prospect 60056
Tel: (312) 593-8230

†Hamilton/Avnet Electronics
3901 No. 25th Ave.
Schiller Park 60176
Tel: (312) 678-6310

INDIANA

Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 547-7777

Sheridan Sales Co.
8790 Purdue Road
Indianapolis 46268
Tel: (317) 297-3146

KANSAS

Hamilton/Avnet Electronics
37 Lenexa Industrial Center
9900 Pflumm Road
Lenexa 66215
Tel: (913) 888-8900

MARYLAND

Cramer/EW Baltimore
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5790

†Cramer/EW Washington
16021 Industrial Drive
Gaithersburg 20760
Tel: (301) 948-0110

†Hamilton/Avnet Electronics
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5000

MASSACHUSETTS

†Cramer Electronics Inc.
85 Wells Avenue
Newton 02159
Tel: (617) 969-7700

†Hamilton/Avnet Electronics
100 E. Commerce Way
Woburn 01801
Tel: (617) 273-2120

MICHIGAN

Sheridan Sales Co.
24543 Indoplex Drive
Farmington Hills 48024
Tel: (313) 477-3800

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 729-8500

†Hamilton/Avnet Electronics
12870 Farmington Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

MINNESOTA

†Industrial Components
5280 West 74th Street
Minneapolis 55435
Tel: (612) 831-2666

Cramer/Bonn
7275 Bush Lake Road
Edina 55435
Tel: (612) 835-7811

†Hamilton/Avnet Electronics
7683 Washington Avenue So.
Edina 55435
Tel: (612) 941-3801

MISSOURI

†Hamilton/Avnet Electronics
364 Brookes Lane
Hazelwood 63042
Tel: (314) 731-1144

NEW JERSEY

Cramer/Pennsylvania, Inc.
12 Springdale Road
Cherry Hill Industrial Center
Cherry Hill 08003
Tel: (609) 424-5993
TWX: 710-896-0908

Components Plus
310 Railroad Avenue
Hackensack 07601
Tel: (201) 487-0565

NEW JERSEY (cont.)

†Hamilton/Avnet Electronics
218 Little Falls Road
Cedar Grove 07009
Tel: (201) 239-0800
TWX: 710-994-5787

Cramer/New Jersey
No. 1 Barrett Avenue
Moonachie 07074
Tel: (201) 935-5600

†Hamilton/Avnet Electronics
113 Gaither Drive
East Gate Industrial Park
Mt. Laurel 08057
Tel: (609) 234-2133
TWX: 710-897-1405

NEW MEXICO

Hamilton/Avnet Electronics
2450 Baylor Drive, S.E.
Albuquerque 87119
Tel: (505) 765-1500

Cramer/New Mexico
137 Vermont, N.E.
Albuquerque 87108
Tel: (505) 265-5767

NEW YORK

Cramer/Rochester
3000 Winton Road South
Rochester 14623
Tel: (716) 275-0300

Components Plus
40 Oser Avenue
Hauppauge 11787
Tel: (516) 231-9200

†Hamilton/Avnet Electronics
167 Clay Road
Rochester 14623
Tel: (716) 442-7820

†Cramer/Syracuse
6716 Joy Road
East Syracuse 13057
Tel: (315) 437-6671

†Hamilton/Avnet Electronics
6500 Joy Road
E. Syracuse 13057
Tel: (315) 437-2642

†Cramer/Long Island
29 Oser Avenue
Hauppauge, L.I. 11787
Tel: (516) 231-5600
TWX: 510-227-9863

†Hamilton/Avnet Electronics
70 State Street
Westbury, L.I. 11590
Tel: (516) 333-5800
TWX: 510-222-8237

NORTH CAROLINA

Cramer Electronics
938 Burke Street
Winston-Salem 27102
Tel: (919) 725-8711

OHIO

†Hamilton/Avnet Electronics
118 Westpark Road
Dayton 45459
Tel: (513) 433-0610
TWX: 810-450-2531

†Pioneer/Dayton
1900 Troy Street
Dayton 45404
Tel: (513) 236-9900

†Sheridan Sales Co.
10 Knollcrest Drive
Cincinnati 45222
Tel: (513) 761-5432
TWX: 810-461-2670

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600

†Hamilton/Avnet Electronics
761 Beta Drive
Cleveland 44143
Tel: (216) 461-1400

Sheridan Sales Co.
23224 Commerce Park Road
Beachwood 44122
Tel: (216) 831-0130

Sheridan Sales Co.
Shiloh Building, Suite 250
5045 North Main Street
Dayton 45405
Tel: (513) 277-8911

OKLAHOMA

Components Specialties, Inc.
7920 E. 40th Street
Tulsa 74145
Tel: (918) 664-2820

OREGON

Almac/Stroum Electronics
4475 S.W. Scholls Ferry Rd.
Portland 97225
Tel: (503) 292-3534

PENNSYLVANIA

Sheridan Sales Co.
1717 Penn Avenue, Suite 5009
Pittsburgh 15221
Tel: (412) 244-1640

Pioneer/Pittsburgh
560 Alpha Drive
Pittsburgh 15238
Tel: (412) 782-2300

TEXAS

Cramer Electronics
13740 Midway Road
Dallas 75240
Tel: (214) 661-9300

†Hamilton/Avnet Electronics
4445 Sigma Road
Dallas 75240
Tel: (214) 661-8661

†Hamilton/Avnet Electronics
1216 W. Clay
Houston 77019
Tel: (713) 526-4661

Component Specialties, Inc.
10907 Shady Trail, Suite 101
Dallas 75220
Tel: (214) 357-4576

†Component Specialties, Inc.
7313 Ashcroft Street
Houston 77036
Tel: (713) 771-7237

UTAH

Cramer/Utah
391 W. 2500 South
Salt Lake City 84115
Tel: (801) 487-4131

Hamilton/Avnet Electronics
647 W. Billinis Road
Salt Lake City 84119
Tel: (801) 262-8451

WASHINGTON

†Hamilton/Avnet Electronics
13407 Northrup Way
Bellevue 98005
Tel: (206) 746-8750

†Almac/Stroum Electronics
5811 Sixth Ave. South
Seattle 98108
Tel: (206) 763-2300

Cramer/Seattle
1059 Andover Park East
Tukwila 98188
Tel: (206) 575-0907

CANADA

ALBERTA

L. A. Varah Ltd.
4742 14th Street N.E.
Calgary T2E 6LT
Tel: (403) 276-8818
Telex: 13 825 89 77

BRITISH COLUMBIA

†L.A. Varah Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068
Telex: 04 53167

ONTARIO

Cramer/Canada
920 Alness Avenue, Unit No. 9
Downsview
Toronto 392 M3J 2H7
Tel: (416) 661-9222
TWX: 610-492-6210

Hamilton/Avnet Electronics
6291-16 Dorman Road
Mississauga L4V 1H2
Tel: (416) 677-7432
TWX: 610-492-8867

Hamilton/Avnet Electronics
1735 Courtwood Cresc.
Ottawa K2C 2B4
Tel: (613) 226-1700
TWX: 610 562-1906

QUEBEC

†Hamilton/Avnet Electronics
2670 Paulus
St. Laurent H4S 1G2
Tel: (514) 331-6443
TWX: 610-421-3731

MANITOBA

L.A. Varah Ltd.
153 Corbett Drive
Winnipeg R2Y 1V4
Tel: (204) 889-9607

†MDS Centers

Printed in U.S.A. MCS-073-0576/30K